

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
PROGRAMA DE PÓS GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UMA TÉCNICA BASEADA EM EVENTOS
PARA RASTREABILIDADE ENTRE
REQUISITOS MODELADOS COM SysML**

Telmo Oliveira de Jesus

São Cristovão
2017

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
PROGRAMA DE PÓS GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Telmo Oliveira de Jesus

**UMA TÉCNICA BASEADA EM EVENTOS PARA
RASTREABILIDADE ENTRE REQUISITOS
MODELADOS COM SysML**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PROCC) da Universidade Federal de Sergipe (UFS) como parte de requisito para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Michel dos Santos Soares

**São Cristovão
2017**

**FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL
UNIVERSIDADE FEDERAL DE SERGIPE**

J58d Jesus, Telmo Oliveira de
Uma técnica baseada em eventos para rastreabilidade entre requisitos modelados com SysML / Telmo Oliveira de Jesus ; orientador Michel dos Santos Soares. – São Cristóvão, 2017.
71 f. : il.

Dissertação (mestrado em Ciências da computação)–
Universidade Federal de Sergipe, 2017.

1. Programas de computador - Testes. 2. Gerenciamento de configurações de software. 3. Software - Testes. I. Soares, Michel dos Santos. II. Título.

CDU 004.4

Telmo Oliveira de Jesus

**UMA TÉCNICA BASEADA EM EVENTOS PARA
RASTREABILIDADE ENTRE REQUISITOS
MODELADOS COM SysML**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PROCC) da Universidade Federal de Sergipe (UFS) como parte de requisito para obtenção do título de Mestre em Ciência da Computação.

BANCA EXAMINADORA

Prof. Dr. Michel dos Santos Soares, Presidente
Universidade Federal de Sergipe (UFS)

Prof. Dra. Leila Maciel de Almeida e Silva, Membro
Universidade Federal de Sergipe (UFS)

Prof. Dr. Heitor Augustus Xavier Costa, Membro
Universidade Federal de Lavras (UFLA)

UMA TÉCNICA BASEADA EM EVENTOS PARA RASTREABILIDADE ENTRE REQUISITOS MODELADOS COM SysML

Este exemplar corresponde a redação da Dissertação de Mestrado, sendo a defesa do mestrando **Telmo Oliveira de Jesus** para ser aprovada pela banca examinadora.

Trabalho aprovado. São Cristovão, 12 de junho de 2017:

Prof. Dr. Michel dos Santos Soares
Orientador

Prof. Dra. Leila Maciel de Almeida e Silva
Membro

Prof. Dr. Heitor Augustus Xavier Costa
Membro

Este trabalho é dedicado aos meus pais, José Barbosa e Ana Lúcia, por sempre incentivar nossa família aos estudos e pelos valores passados a todos os seus filhos.

Agradecimentos

Agradeço aos meus pais e irmãos pelo incentivo dado durante todo o mestrado.

À minha esposa, Fernanda, e meus filhos, Letícia e Mateus pela compreensão, carinho e apoio em todos os momentos.

Ao meu orientador, Michel, pelos ensinamentos, sabedoria e paciência.

Aos colegas de mestrado pelos momentos de descontração, pela amizade e parcerias criadas.

Aos professores do PROCC, pelo aprendizado.

Aos avaliadores da ferramenta REQTRACE por disponibilizar o tempo deles para a avaliação da ferramenta.

À agência de fomento de pesquisa CNPq (projeto 445500/2014-0).

Resumo

Gerenciar um projeto de software é uma tarefa cada vez mais complexa à medida que as exigências sobre a qualidade do produto final aumentam. Assim, é preciso um gerenciamento eficaz desde a concepção do projeto até a implantação e manutenção do software. Segundo a literatura encontrada, a Engenharia de Requisitos é a fase que possui maior impacto em um projeto de software. As mudanças nos requisitos de software ocorrem ao longo do processo de software, desde a elicitação e análise de requisitos até a operação do software. A rastreabilidade de requisitos possibilita identificar a origem e a dependência entre os requisitos de software. Estudos mostram que as ferramentas e métodos de rastreabilidade de requisitos atuais são insuficientes e dificultam o uso prático da rastreabilidade. O objetivo deste trabalho é a criação de uma técnica de rastreabilidade de requisitos combinada com uma linguagem de modelagem e que seja apoiada por uma ferramenta. Para atingir este objetivo, foram utilizados como instrumentos de pesquisa a revisão bibliográfica, uma pesquisa do tipo estado da arte, análise comparativa, questionário e entrevistas. Inicialmente, foram realizadas análise das técnicas de rastreabilidade de requisitos e avaliação de ferramentas de rastreabilidade de requisitos com intuito de avaliar se as técnicas são utilizadas na prática, e se estão sendo apoiadas por ferramentas de software. Após estas análises, foi proposta uma técnica de rastreabilidade entre requisitos baseada em eventos utilizando conceitos da linguagem de modelagem SysML. Uma ferramenta de software foi desenvolvida para apoiar essa técnica. Sempre que um requisito é modificado, a ferramenta notifica aos *stakeholders* a alteração realizada nos requisitos do sistema por meio de *e-mails*. A técnica e a ferramenta foram avaliadas por vinte e quatro profissionais de tecnologia da informação utilizando questionário, baseado no modelo TAM, que engloba os conceitos de utilidade percebida, facilidade de uso percebido e uso percebido, e entrevistas. Como resultado, foram observados a falta de relação entre as técnicas e as ferramentas pesquisadas e que o critério de comunicação automática com *stakeholders* não foi abordado em nenhuma ferramenta avaliada. Com relação à técnica e ferramenta propostas, observou-se uma ampla aceitação da técnica e da ferramenta. A aceitação das propostas foi realizada por profissionais com experiências variadas em desenvolvimento de software.

Palavras-chaves: Gerenciamento de Requisitos, Rastreabilidade de Requisitos, Técnicas de Rastreabilidade, Ferramentas de Rastreabilidade, SysML, TAM.

Abstract

Managing a software project is an increasingly complex task as it demands the quality of the final product to increase. Thus, effective management is required from project design to software deployment and maintenance. According to the literature found, Requirements Engineering is the most important phase in terms of final impact on a software project. Changes in software requirements occur throughout the software process, from elicitation and requirements analysis to software operation. Requirements traceability enables one to identify the source and the dependency among software requirements. Studies show that current requirements traceability tools and methods are insufficient which makes it difficult to use traceability in practice. The goal of this work was the proposal of a requirements traceability technique combined with a modeling language and supported by a tool. To achieve this goal, bibliographic review, state-of-the-art research, comparative analysis, questionnaire and interviews were used as research instruments. Initially, an analysis of requirements traceability techniques and evaluation of requirements traceability tools were performed to evaluate if the techniques are used in practice and are being supported by software tools. After these analysis, an event-based traceability technique was proposed using concepts from the SysML modeling language. A software tool was developed to support this technique. Whenever a requirement is modified, the tool notifies stakeholders of such change made to the system requirements through e-mail. The technique and the tool were evaluated by twenty four information technology professionals using a questionnaire based on the TAM model that encompasses the concepts of perceived utility, perceived ease of use and perceived use, and interviews. As a result, the lack of relationship between the techniques and the tools surveyed was observed, and the criterion of automatic communication with stakeholders was not addressed in any evaluated tool. With regard to the proposed technique and tool, a wide acceptance of technique and tool was observed. The acceptance of the proposals was carried out by professionals with varied experiences in software development.

Key-words: Requirements Management, Requirements Traceability, Traceability Techniques, Traceability Tools, SysML, TAM.

Lista de figuras

Figura 2.1 – Rastreabilidade <i>Backward</i> e <i>Forward</i>	24
Figura 2.2 – Rastreabilidade Pré-Requisitos e Pós-Requisitos	25
Figura 2.3 – Diagrama de Caso de Uso em UML	27
Figura 2.4 – Representação de um requisito - Diagrama de Requisitos em SysML	29
Figura 2.5 – Diagrama de Requisitos em SysML de um Sistema Comercial	30
Figura 2.6 – Diagrama de Requisitos em SysML de um Sistema de Segurança	31
Figura 3.1 – Rastreabilidade baseada em valor	36
Figura 4.1 – Proposta da técnica de rastreabilidade entre requisitos	45
Figura 4.2 – Tela de opções do <i>menu</i> cadastro	46
Figura 4.3 – Tela de opções do <i>menu</i> consultas	47
Figura 4.4 – Diagrama de Caso de Uso da UML da ferramenta REQTRACE	48
Figura 4.5 – Diagrama de Classes da UML da ferramenta REQTRACE	48
Figura 4.6 – Cadastro de <i>stakeholders</i>	49
Figura 4.7 – Cadastro de requisitos	50
Figura 4.8 – <i>E-mail</i> gerado pela ferramenta REQTRACE na inclusão de um requisito	50
Figura 4.9 – <i>E-mail</i> gerado pela ferramenta REQTRACE na modificação de um requisito	50
Figura 4.10– <i>E-mail</i> gerado pela ferramenta REQTRACE na exclusão de um requisito	51
Figura 4.11–Cadastro das relações entre os requisitos	51
Figura 4.12–Consulta de requisitos	52
Figura 4.13–Consulta do histórico de mudanças nos requisitos	52

Lista de tabelas

Tabela 2.1 – Tabela de requisitos em SysML	31
Tabela 2.2 – Tabela de relações entre requisitos em SysML	31
Tabela 3.1 – Matriz de rastreabilidade	35
Tabela 3.2 – Comparativo entre técnicas de rastreabilidade de requisitos	37
Tabela 3.3 – Resumo da avaliação de ferramentas de rastreabilidade de requisitos	40
Tabela 4.1 – Modelos de <i>e-mails</i> (notificações aos <i>stakeholders</i>)	44
Tabela 5.1 – <i>Perceived usefulness</i> (PU) - Sentença 1 até 7 (N=24)	56
Tabela 5.2 – <i>Perceived ease of use</i> (PEOU) - Sentença 1 até 4 (N=24)	56
Tabela 5.3 – <i>Perceived usage</i> (PUE) - Sentença 1 até 4 (N=24)	57
Tabela 5.4 – Correlação entre as hipóteses	57
Tabela 5.5 – Dados coletados na entrevista	60

Lista de abreviaturas e siglas

CASE	Computer-Aided Software Engineering
EA	Estado da Arte
EBT	Event Based Traceability
ER	Engenharia de Requisitos
ERP	Enterprise Resource Planning
ES	Engenharia de Software
PEOU	Perceived Ease of Use
PU	Perceived Usefulness
PUE	Perceived Usage
OMG	Object Management Group
SMTP	Simple Mail Transfer Protocol
SSL	Secure Sockets Layer
SysML	Systems Modeling Language
TAM	Technology Acceptance Model
UML	Unified Modeling Language
URML	Unified Requirements Modeling Language
XML	Extensible Markup Language
XP	Extreme Programming

Sumário

1	INTRODUÇÃO	14
1.1	Contextualização	14
1.2	Objetivos	15
1.3	Metodologia	16
1.4	Trabalhos Relacionados	17
1.5	Estrutura da Dissertação	19
2	REFERENCIAL TEÓRICO	21
2.1	Requisitos de Software	21
2.2	Engenharia de Requisitos	21
2.3	Gerenciamento de Requisitos	22
2.4	Rastreabilidade de Requisitos	24
2.5	Linguagens de Modelagem de Requisitos	26
2.6	<i>Systems Modeling Language</i> (SysML)	28
2.6.1	Diagrama de Requisitos	28
2.6.2	Tabelas SysML	30
3	TÉCNICAS E FERRAMENTAS DE RASTREABILIDADE DE REQUISITOS	32
3.1	Técnicas de Rastreabilidade de Requisitos	33
3.2	Avaliação das Técnicas Pesquisadas	36
3.3	Ferramentas de Rastreabilidade de Requisitos	37
3.4	CrITÉRIOS de Avaliação das Ferramentas	38
3.5	Avaliação das Ferramentas Pesquisadas	40
3.6	Conclusão do Capítulo	41
4	PROPOSTA DE UMA TÉCNICA E FERRAMENTA DE RASTREABILIDADE ENTRE REQUISITOS	43
4.1	Proposta da Técnica de Rastreabilidade entre Requisitos	43
4.2	Funcionalidades da Ferramenta REQTRACE	45
4.3	Telas da Ferramenta REQTRACE	49
4.4	Conclusão do Capítulo	52
5	AValiaÇÃO DA TÉCNICA E DA FERRAMENTA REQTRACE	54
5.1	Modelo TAM	54
5.2	Avaliação Quantitativa - Questionário	55

5.3	Avaliação Qualitativa - Entrevistas	58
5.4	Conclusão do Capítulo	61
6	CONCLUSÃO	62
6.1	Contribuições do Trabalho	63
6.2	Ameaças à Validade	64
6.3	Limitações da Pesquisa	64
6.4	Trabalhos Futuros	65
	REFERÊNCIAS	66

1 Introdução

Neste capítulo é mostrada uma visão preliminar dos assuntos abordados na dissertação. As seções são apresentadas, nesta ordem: inicialmente, a contextualização na Seção 1.1, em seguida os objetivos do trabalho na Seção 1.2, a metodologia aplicada na Seção 1.3, os trabalhos relacionados na Seção 1.4, e, por fim, a estrutura do trabalho na Seção 1.5.

1.1 Contextualização

Requisitos de um software são descrições dos serviços que devem ser fornecidos por esse software e as suas restrições operacionais (SOMMERVILLE, 2015). Os requisitos de um software incluem especificações das funcionalidades que o software deve fornecer, limites sob os quais ele deve operar, propriedades gerais do software e restrições que devem ser satisfeitas no seu processo de desenvolvimento. O processo de identificar, documentar, analisar e verificar as restrições e serviços de um software é chamado de Engenharia de Requisitos - ER (SOMMERVILLE, 2015).

A ER é uma das principais áreas da Engenharia de Software (ES), sendo uma das primeiras etapas para o desenvolvimento do software (RAHMAN; RIPON, 2014). A ER pode ser dividida em dois grupos de atividades: (i) o desenvolvimento de requisitos, que inclui as atividades de como elicitar, documentar, analisar e validar os requisitos; (ii) e o gerenciamento de requisitos, que inclui atividades relacionadas à manutenção, tais como rastreabilidade e gerenciamento de mudanças de requisitos (PARVIAINEN *et al.*, 2005).

Uma das atividades relativas ao gerenciamento de requisitos é a rastreabilidade de requisitos que pode ser definida como a habilidade de descrever e acompanhar um requisito no ciclo de vida do software (HAMMER *et al.*, 1997). A rastreabilidade de requisitos é um mecanismo para a gestão e auditoria de todo o processo de desenvolvimento de software (LAGO; MUCCINI; VLIET, 2009). A rastreabilidade influencia na qualidade de produtos de software facilitando sua reutilização (CLELAND-HUANG; GOTEL; ZISMAN, 2012). A prática efetiva de rastreabilidade ajuda na compreensão do software, análise de impacto, *debug* do software e comunicação entre os membros da equipe (ASUNCION; FRANCOIS; TAYLOR, 2007). A utilização de ferramentas ajuda a aprimorar a precisão e reduz o tempo necessário para identificar relações de rastreabilidade (LUCIA *et al.*, 2007) obtendo benefícios econômicos, técnicos e sociais (CYSNEIROS FILHO, 2011).

Desenvolver um software considerando efetivamente a rastreabilidade completa e consistente entre os artefatos produzidos ainda é um grande desafio na prática (BIRÓ *et al.*, 2016). A ausência de rastreabilidade torna difícil a análise de impacto quando há mudanças nos requisitos (MIN, 2016). Kannenberg e Saiedian (2009) citam os principais desafios na implantação

da rastreabilidade, como alto custo, grande esforço de tempo, dificuldade de manutenção da rastreabilidade em virtude das mudanças no software, problemas organizacionais e utilização de métodos manuais.

Este trabalho tem como propósito a criação de uma nova técnica de rastreabilidade entre requisitos usando uma linguagem com um diagrama específico de modelagem de requisitos, apoiada por uma ferramenta desenvolvida durante a pesquisa.

1.2 Objetivos

Diante da importância em considerar a rastreabilidade de requisitos como atividade essencial da ER, este trabalho tem como objetivo principal desenvolver uma técnica e uma ferramenta de rastreabilidade entre requisitos.

Este trabalho propõe uma técnica de rastreabilidade entre requisitos combinada com conceitos da linguagem de modelagem *Systems Modeling Language* (SysML) com o intuito de solucionar problemas e desafios mencionados no trabalho de Regan *et al.* (2012), como falta de comunicação entre as equipes e dificuldades na utilização de ferramentas. A escolha da SysML deve-se à flexibilidade de representar diversos tipos de requisitos (inclusive requisitos não funcionais), permitir melhor comunicação com *stakeholders* por meio do diagrama de Requisitos, além de mostrar os vários tipos de relacionamentos entre diferentes requisitos. A ferramenta é usada como suporte para a técnica proposta. A junção de uma técnica e de uma linguagem de modelagem apoiadas por uma ferramenta não foi observada nas diversas pesquisas realizadas durante este trabalho conforme apresentado na Seção 1.4.

Para atingir o objetivo principal, foram estabelecidos os seguintes objetivos específicos:

1. Identificar, analisar e comparar técnicas e ferramentas de rastreabilidade de requisitos existentes. Este objetivo específico é tratado no Capítulo 3.
2. Definir critérios baseados em estudos da literatura para avaliação das ferramentas pesquisadas. Este objetivo específico é tratado no Capítulo 3.
3. Propor uma técnica de rastreabilidade entre requisitos modelados usando SysML. A linguagem de modelagem SysML é discutida na Seção 2.6. Uma ferramenta é implementada para apoiar a técnica proposta. Este objetivo específico é tratado no Capítulo 4.
4. Avaliar a técnica e a ferramenta implementada, na prática, com profissionais da área. Este objetivo específico é tratado no Capítulo 5.

1.3 Metodologia

Shaw (2003) menciona que existem vários tipos de pesquisa e perguntas intimamente relacionadas a questões de ES e destaca que uma pesquisa em ES deve conter de forma clara o tipo de pesquisa, o resultado de pesquisa e a validação da pesquisa. Com relação a esta classificação, este trabalho utilizou o tipo de pesquisa “método de análise ou avaliação”, como resultado de pesquisa uma “técnica” e uma “ferramenta” e como validação da pesquisa uma “avaliação”.

Segundo Shaw (2003), uma técnica refere-se à uma nova ou melhor maneira de fazer alguma tarefa, como projeto, implementação, manutenção, medição ou avaliação. Uma técnica deve ser operacional - não conselhos ou diretrizes. A proposta deste trabalho refere-se a uma técnica de rastreabilidade entre requisitos.

Para que os objetivos deste trabalho fossem atingidos, quatro instrumentos de pesquisa foram aplicados: revisão bibliográfica (incluindo pesquisa do tipo estado da arte), análise comparativa, questionário e entrevistas.

Inicialmente, foi realizada uma revisão bibliográfica com o objetivo de encontrar trabalhos relevantes que pudessem auxiliar no desenvolvimento de uma técnica de rastreabilidade entre requisitos apoiada por uma ferramenta. Esta revisão é altamente recomendável para obter conhecimento sobre um tema fornecendo direcionamentos úteis que colaboram para os resultados de uma pesquisa (SJOBERG; DYBA; JORGENSEN, 2007). Esta revisão bibliográfica é descrita no Capítulo 2.

Uma pesquisa do tipo estado da arte (EA) também foi realizada. O EA tem como objetivo a sistematização da produção em determinada área de conhecimento, fundamental para apreender a amplitude do que vem sendo produzido. O EA não se restringe à identificação da produção, mas também analisá-la e categorizá-la revelando os múltiplos enfoques e perspectivas (ROMANOWSKI; ENS, 2006). Os artigos científicos foram consultados nas bibliotecas digitais *ACM*, *IEEEExplore*, *ScienceDirect* e *Scopus* nos últimos dez anos escritos na língua inglesa. Os trabalhos mais relevantes são apresentados na Seção 1.4.

Após a pesquisa do EA, foram realizadas pesquisas aplicadas e qualitativas para encontrar ferramentas *open source* de rastreabilidade de requisitos com o intuito de avaliar as suas funcionalidades. As funcionalidades foram categorizadas e os critérios de avaliação foram definidos. A pesquisa é apresentada na Seção 3.3 e a definição dos critérios é descrita na Seção 3.4. Em seguida, foi realizada uma análise comparativa das técnicas e das ferramentas de rastreabilidade de requisitos. A análise comparativa tem como objetivo a investigação de indivíduos, fenômenos ou fatos, com vistas a ressaltar as diferenças e similaridades entre eles. Segundo Fachin (2005), o método comparativo consiste em investigar coisas ou fatos e explicá-los segundo suas semelhanças e suas diferenças, além de permitir a análise de dados concretos e a dedução de semelhanças e divergências de elementos constantes, abstratos e gerais, propiciando investigações de caráter

indireto. A análise comparativa das técnicas é apresentada na Seção 3.2 e das ferramentas na Seção 3.5.

Após as análises comparativas, uma técnica de rastreabilidade entre requisitos foi proposta neste trabalho com o intuito de melhorar a comunicação com os *stakeholders*. Para apoiar a técnica proposta, foi implementada uma ferramenta denominada REQTRACE. A técnica, as funcionalidades e as telas da ferramenta são apresentadas no Capítulo 4.

Com o objetivo de avaliar as propostas deste trabalho, foram realizadas uma avaliação quantitativa baseada em questionário e uma avaliação qualitativa por meio de entrevistas. O questionário e as entrevistas foram realizados com profissionais da área de computação, professores e alunos de mestrado.

O questionário é uma técnica de investigação composta por um conjunto de questões que são submetidas a pessoas com o propósito de obter informações sobre conhecimentos, crenças, valores ou interesses. Com relação ao formato das questões, este trabalho utilizou questões fechadas. Neste tipo de questão, pede-se aos respondentes para que escolham uma alternativa dentre as que são apresentadas numa lista permitindo maior uniformidade às respostas além de poder ser facilmente processadas (GIL, 2008). O questionário é apresentado na Seção 5.2.

A entrevista é uma técnica em que o entrevistador se apresenta frente ao investigado e lhe formula perguntas, com o objetivo de obtenção de dados que interessam à investigação. A entrevista é adequada para a obtenção de informações acerca do que as pessoas sabem, creem, esperam, sentem ou desejam, pretendem fazer, fazem ou fizeram, bem como acerca das suas explicações ou razões a respeito das coisas precedentes. Por sua flexibilidade, é adotada como uma técnica fundamental de investigação nos mais diversos campos. Com relação ao nível de estruturação, foi utilizada a entrevista estruturada cujo propósito desenvolve-se a partir de uma relação fixa de perguntas, cuja ordem e redação permanece invariável para todos os entrevistados além de permitir uma análise estatística dos dados já que as respostas obtidas são padronizadas (GIL, 2008). O detalhamento das entrevistas é apresentado na Seção 5.3.

Considerando os fatores que motivaram este trabalho, destacam-se as dificuldades associadas à rastreabilidade entre requisitos e o baixo número de técnicas de rastreabilidade entre requisitos apoiadas por uma ferramenta. Desta forma, a motivação desta dissertação foi contribuir para melhorar o monitoramento entre os requisitos de um software buscando formas de solucionar por meio de uma técnica apoiada por uma ferramenta.

1.4 Trabalhos Relacionados

A ER busca minimizar as dificuldades dos engenheiros de software para identificar os requisitos do software a ser desenvolvido por meio de atividades que levam a um entendimento do negócio, das necessidades do cliente e de como será a interação do usuário com o software

(PRESSMAN, 2016).

O gerenciamento de requisitos inclui atividades que auxiliam a equipe a controlar, identificar, rastrear e gerenciar mudanças durante o ciclo de vida do software.

A rastreabilidade de requisitos é definida como as relações existentes entre os artefatos resultantes do processo de desenvolvimento de software. Essa etapa está relacionada a manter ligações entre os artefatos, isto é, fornecer meios para determinar as relações e dependências entre os artefatos que apoiam atividades da ES, como o impacto das alterações, análise e manutenção.

Diversas pesquisas têm sido realizadas na área de rastreabilidade de requisitos em temas como técnicas de rastreabilidade (TORKAR *et al.*, 2012), (EDER *et al.*, 2015), modelos de rastreabilidade (BADREDDIN; STURM; LETHBRIDGE, 2014), (OH; KANG, 2014), (PLATANOTIS *et al.*, 2015) e automatização da rastreabilidade (THOMMAZO *et al.*, 2012), (EDER *et al.*, 2015).

Nair, Vara e Sen (2013) realizaram uma análise das publicações em conferências de ER nos últimos vinte anos sobre rastreabilidade. Eles concluem que há grande foco na rastreabilidade pós-requisitos (refere-se aos aspectos da vida de um requisito que resultam da inclusão na especificação de requisitos) e mencionam os principais desafios da rastreabilidade: falta de conhecimento e de entendimento do tema (ARKLEY; RIDDLE, 2005) e a manutenção da rastreabilidade em virtude da evolução dos requisitos (CHARRADA; KOZIOLEK; GLINZ, 2012), (NIU; MAHMOUD, 2012). A maioria das contribuições dos artigos foram técnicas com abordagens e metodologias com apoio de ferramentas, incluindo uma gama de características em ferramentas de rastreabilidade como rastreamento de todo o ciclo de vida do artefato (criação e atualização da rastreabilidade entre diversos artefatos), manutenção da rastreabilidade entre artefatos específicos para especificação de requisitos e automatização da rastreabilidade. Os autores destacam que nos últimos anos há interesse crescente nos temas de automatização da rastreabilidade, modelo de rastreabilidade, qualidade da rastreabilidade, experimentação e colaboração entre academia e indústria. No que diz respeito às áreas para novas pesquisas, os autores consideram que a visualização da rastreabilidade, análise de impacto e ferramentas devem ser estudadas com mais profundidade.

Regan *et al.* (2012) analisaram artigos entre 2004 e 2012 que tratavam de implementação de software de rastreabilidade e problemas/barreiras para implementar rastreabilidade. Os autores classificaram tais barreiras nos estudos de casos selecionados em três categorias apontando soluções para cada barreira identificada. As categorias foram divididas em problemas de gerenciamento, problemas sociais e problemas técnicos. Custos associados à implementação da rastreabilidade, poucas métricas para calcular o retorno sobre o investimento e falta de orientação dos profissionais para realização da rastreabilidade são exemplos dos problemas de gerenciamento. Com relação aos problemas sociais, são citadas as barreiras das diferentes visões dos *stakeholders*, políticas internas e falta de comunicação entre as equipes. Nos problemas técnicos são mencionados as dificuldades com as ferramentas, armazenamento e controle de

versão das relações de rastreabilidade e complexidade (por exemplo, grande número de artefatos e de rastreabilidade de requisitos não funcionais). Entre algumas soluções propostas para as barreiras, pode-se citar gestão da educação, criação de políticas organizacionais, integração de tarefas de rastreabilidade nas práticas de trabalho, criação de um repositório de artefatos integrado e utilização de técnicas e ferramentas. Ao final, os autores sugerem colocar em prática na indústria as soluções propostas e realizar um novo estudo sobre a eficácia dessas soluções.

Chang *et al.* (2014) propõem o uso da SysML no modelo de elicitação de requisitos para coletar e modelar os requisitos do usuário e transformar o diagrama de Requisitos da SysML em outros diagramas para projeto e análise de acordo com as regras de transformação definidas pela linguagem de transformação ATLAS. Esta abordagem visa fornecer um procedimento e uma ferramenta com interface gráfica para elicitar, analisar, validar e avaliar os requisitos do sistema garantindo que o software satisfaça as necessidades dos usuários. A linguagem de transformação ATLAS define regras de transformação entre diagramas SysML e *Unified Modeling Language* (UML) fornecendo maneiras de produzir um conjunto de modelos alvos a partir de um conjunto de modelos.

Após análise destes trabalhos, conclui-se que houve pouco foco na rastreabilidade entre requisitos, ausência de utilização de técnicas apoiadas por ferramentas e ausência de utilização das relações e tabelas SysML. Desta forma, identificou-se a necessidade de mais pesquisas na área de rastreabilidade entre requisitos e de maior utilização de técnicas e de ferramentas além de melhorar a comunicação entre as equipes. Uma opção para solucionar estes problemas é a utilização ou a criação de novas técnicas de rastreabilidade entre requisitos combinadas com conceitos de uma linguagem de modelagem (SysML). A implementação de uma ferramenta apoia o uso prático da rastreabilidade entre requisitos. Este trabalho combina uma técnica, a linguagem de modelagem SysML (relações e tabelas SysML) e uma ferramenta, com o propósito de documentar, analisar e manter a rastreabilidade entre requisitos.

1.5 Estrutura da Dissertação

Este trabalho contribui para a pesquisa e a prática em Engenharia de Software, especialmente para a subárea Engenharia de Requisitos. O texto desta dissertação está organizado em seis capítulos. Uma breve descrição do conteúdo de cada capítulo é apresentada a seguir.

No Capítulo 2 são apresentados conteúdos relevantes para a realização deste trabalho. É descrita a fundamentação teórica sobre a pesquisa proposta descrevendo conceitos de requisitos de software, Engenharia de Requisitos, gerenciamento de requisitos, rastreabilidade de requisitos, linguagens de modelagem de requisitos e SysML.

No Capítulo 3 são destacadas as técnicas e as ferramentas de rastreabilidade de requisitos pesquisadas além da definição dos critérios utilizados para sua avaliação. Este capítulo é finalizado com uma análise comparativa das técnicas e das ferramentas pesquisadas.

Uma proposta de técnica e uma ferramenta denominada REQTRACE são descritas no Capítulo 4.

Uma avaliação quantitativa, utilizando o modelo *Technology Acceptance Model* (TAM), e uma qualitativa são apresentadas no Capítulo 5.

Por fim, no Capítulo 6 são apresentadas as conclusões referente ao trabalho, as principais contribuições, as ameaças à validade e propostas para pesquisas futuras.

2 Referencial Teórico

2.1 Requisitos de Software

Requisitos de um software são descrições dos serviços que devem ser fornecidos por esse software e as suas restrições operacionais (SOMMERVILLE, 2015). Pode-se dizer que os requisitos de um software incluem especificações das funcionalidades que o software deve fornecer, limites sob as quais ele deve operar, propriedades gerais do software e restrições que devem ser satisfeitas no seu processo de desenvolvimento.

Uma classificação amplamente aceita na comunidade científica quanto ao tipo de informação documentada por um requisito faz a distinção entre requisitos funcionais e requisitos não funcionais.

- Requisitos funcionais são declarações de serviços que o software deve prover, descrevendo o que o software deve fazer (SOMMERVILLE, 2015). Um requisito funcional descreve uma interação entre o software e o seu ambiente (PFLEEGER, 2004), podendo descrever, ainda, como o software deve reagir a entradas específicas, como o software deve se comportar em situações específicas e o que o software não deve fazer (SOMMERVILLE, 2015).
- Requisitos não funcionais descrevem restrições sobre os serviços ou funções oferecidos pelo software (SOMMERVILLE, 2015), as quais limitam as opções para criar uma solução para o problema (PFLEEGER, 2004). Neste sentido, os requisitos não funcionais são muito importantes para a fase de projeto, servindo como base para a tomada de decisões nessa fase. Os requisitos não funcionais têm origem nas necessidades dos usuários, em restrições de orçamento, em políticas organizacionais, em necessidades de interoperabilidade com outros sistemas de software ou hardware ou em fatores externos como regulamentos e legislações (SOMMERVILLE, 2015).

2.2 Engenharia de Requisitos

O processo de identificar, documentar, analisar e verificar as restrições e serviços de um software é chamado de Engenharia de Requisitos (SOMMERVILLE, 2015). O processo de ER envolve criatividade, interação de diferentes pessoas, conhecimento e experiência para transformar informações diversas (sobre a organização e leis sobre o software a ser construído) em documentos e modelos que direcionem o desenvolvimento de software (SOMMERVILLE; KOTONYA, 1998).

Processos de ER podem variar de uma organização para outra, ou até mesmo dentro de uma organização específica, em função de características dos projetos. A definição de um processo apropriado à organização traz muitos benefícios, pois uma descrição desse processo fornecerá orientações e reduzirá a probabilidade de esquecimento ou de uma execução superficial. Alguns benefícios que um processo de ER de alta qualidade pode trazer são: menor quantidade de defeitos nos requisitos, redução de retrabalho, desenvolvimento de menos características desnecessárias, diminuição de custos, desenvolvimento mais rápido, menos problemas de comunicação, alterações de escopo reduzidas, estimativas mais confiáveis e maior satisfação dos clientes e membros da equipe (WIEGERS, 2003).

Ainda que diferentes projetos requeiram processos com características específicas para contemplar suas peculiaridades, é possível estabelecer um conjunto de atividades básicas que deve ser considerado na definição de um processo de ER. Tomando por base o processo proposto por Sommerville e Kotonya (1998), neste texto considera-se que um processo de ER deve contemplar, tipicamente, as atividades de levantamento de requisitos, análise de requisitos, documentação de requisitos, verificação e validação de requisitos e gerência de requisitos.

O processo começa pelo levantamento de requisitos, que deve considerar necessidades dos usuários e clientes, informações de domínio, conjunto de software existente, regulamentos e leis. Uma vez identificados os requisitos, é possível iniciar a atividade de análise, quando os requisitos levantados são usados como base para a modelagem dos requisitos. Tanto no levantamento, quanto na análise de requisitos, é importante documentar requisitos e modelos. Os documentos produzidos são, então, verificados e validados. Adicionalmente, um esforço de garantia da qualidade deve ser realizado, visando garantir conformidade em relação a padrões e ao processo estabelecidos pela organização. Caso clientes, usuários e desenvolvedores estejam de acordo com os requisitos, o processo de desenvolvimento pode avançar; caso contrário, deve-se retornar à atividade correspondente para resolver os problemas identificados. Em paralelo às atividades anteriormente mencionadas, há a gerência de requisitos, que se ocupa em gerenciar mudanças nos requisitos.

Vale destacar que não há limite bem definido entre as atividades citadas. Na prática, elas são intercaladas e existe alto grau de iteração e *feedback* entre elas. O processo é executado até os usuários estarem satisfeitos e concordarem com os requisitos (SOMMERVILLE; KOTONYA, 1998).

2.3 Gerenciamento de Requisitos

Mudanças nos requisitos ocorrem ao longo de todo o processo de desenvolvimento de software, desde o levantamento e análise de requisitos até durante a operação do software. Elas são decorrentes de diversos fatores, tais como, descoberta de erros, omissões, conflitos e inconsistências nos requisitos, melhor entendimento por parte dos usuários de suas necessidades,

problemas técnicos, de cronograma ou de custo, mudança nas prioridades do cliente, mudanças no negócio, aparecimento de novos competidores, mudanças econômicas, mudanças na equipe, mudanças no ambiente onde o software será instalado e mudanças organizacionais ou legais. Para minimizar as dificuldades impostas por essas mudanças, é necessário gerenciar requisitos (TOGNERI; FALBO; MENEZES, 2002).

O processo de gerência de requisitos envolve as atividades que ajudam a equipe de desenvolvimento a identificar, controlar e rastrear requisitos e gerenciar mudanças de requisitos em qualquer momento ao longo do ciclo de vida do software (SOMMERVILLE; KOTONYA, 1998; PRESSMAN, 2016). Os principais objetivos desse processo são: gerenciar alterações nos requisitos acordados; gerenciar relacionamentos entre requisitos e gerenciar dependências entre requisitos e outros documentos produzidos durante o processo de software (SOMMERVILLE; KOTONYA, 1998). Para tal, o processo de gerência de requisitos inclui diversas atividades, como controle de mudanças, controle de versão, acompanhamento do estado dos requisitos e rastreamento de requisitos.

O controle de mudança define os procedimentos, processos e padrões que devem ser utilizados para gerenciar as alterações de requisitos, assegurando que qualquer proposta de mudança seja analisada conforme os critérios estabelecidos pela organização. Mudanças podem ser necessárias em diferentes momentos e por diferentes razões. Para garantir uma abordagem consistente, recomenda-se que as organizações definam um conjunto de políticas de gerência de mudança (SOMMERVILLE; KOTONYA, 1998). Se as mudanças não forem controladas, alterações com baixa prioridade podem ser implementadas antes de outras mais importantes e modificações com custo alto que não são realmente necessárias podem ser aprovadas (TOGNERI; FALBO; MENEZES, 2002).

Yu e Shen (2013) destacam os principais problemas associados ao gerenciamento de requisitos baseados em entrevistas semi-estruturadas e em quatro estudos de casos. Os principais problemas envolvem *stakeholders* inexperientes, identificação e representação inadequadas dos requisitos durante o processo de desenvolvimento, abordagem não estruturada, falta de compreensão e erros na interpretação das necessidades e exigências dos *stakeholders*, tempo insuficiente para elaborar uma estrutura adequada para o gerenciamento de requisitos e necessidades inadequadas de esforço ao longo do ciclo de vida do *software*. Como possíveis soluções para os problemas mencionados, Yu e Shen (2013) sugerem designar um participante experiente do projeto como gerente responsável pela gestão dos requisitos do cliente, fornecer procedimentos para registrar, gerenciar e rastrear alterações nos requisitos do cliente e um plano de requisitos para melhorar a prática de gestão de requisitos.

2.4 Rastreabilidade de Requisitos

A rastreabilidade de requisitos pode ser definida como a habilidade de descrever e acompanhar um requisito no ciclo de vida do software (HAMMER *et al.*, 1997). A rastreabilidade fornece uma base para o desenvolvimento de uma trilha de auditoria para todo o projeto, possibilitando encontrar outros requisitos e artefatos que podem ser afetados pelas mudanças solicitadas (THAYER; DORFMAN, 1997). Para tal, é necessário haver ligações entre requisitos e entre requisitos e outros elementos do processo de software. Assim, a identificação da composição de requisitos, das dependências entre requisitos, de requisitos conflitantes, da origem dos requisitos e de seus interessados, além da identificação de em quais artefatos produzidos durante o desenvolvimento de software um requisito é tratado, é de fundamental importância para que a rastreabilidade possa ser implementada (SOMMERVILLE; KOTONYA, 1998), (WIEGERS, 2003), (ROBERTSON; ROBERTSON, 2006).

Existem várias maneiras pelas quais a rastreabilidade pode ser realizada. No que diz respeito à direção de rastreamento, um requisito pode ser rastreado para trás ou para frente; em relação à evolução dos requisitos, estes podem ser rastreados antes ou depois de sua inclusão na especificação de requisitos e no tocante aos objetos envolvidos, pode-se ter rastreabilidade inter-requisitos e extra-requisitos (PINHEIRO, 2004).

Os conceitos de rastreabilidade para trás e para frente tem as seguintes definições:

- Rastreabilidade para trás (*backward*) é a habilidade de rastrear um requisito até a sua fonte, ou seja, uma pessoa, instituição, legislação, entre outros.
- Rastreabilidade para frente (*forward*) é a habilidade de rastrear um requisito em direção a um componente de *design* ou caso de teste.

A Figura 2.1 apresenta as rastreabilidades para trás e para frente e suas direções com relação aos requisitos originais, a especificação de requisitos e outros artefatos de software.

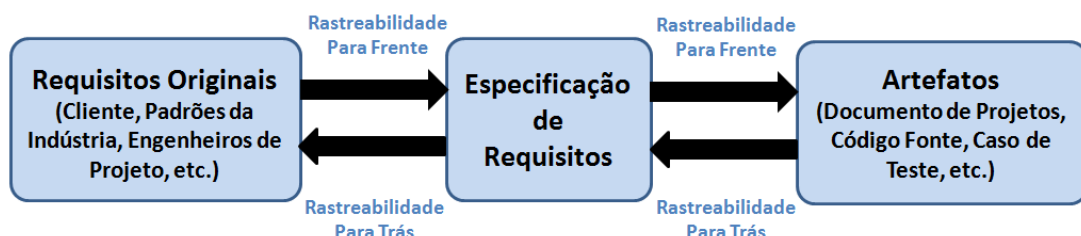


Figura 2.1 – Rastreabilidade *Backward* e *Forward*. Adaptado de (KANNENBERG; SAIEDIAN, 2009)

O requisito é rastreado para trás, por exemplo, quando há uma mudança e existe a necessidade de entendê-la, investigando as informações utilizadas para elicitar o requisito

mencionado. O requisito é rastreado para frente, por exemplo, quando o mesmo vai ser alterado e é relevante a investigação do impacto da mudança, tais como procedimentos de teste relacionados, os componentes construídos para atender tal requisito e mudanças que devem ser realizadas nestes (PINHEIRO, 2004).

A especificação de requisitos é o resultado do processo de elicitação. O rastreamento de um requisito pode ser feito para obter informações relacionadas com o processo de elicitação, antes de sua inclusão na especificação de requisitos, ou para obter informações relacionadas com o seu uso, após o requisito ter sido elicitado e incluído na especificação de requisitos (PINHEIRO, 2004).

A rastreabilidade pré-requisitos refere-se aos aspectos da vida de um requisito prévio à sua inclusão na especificação de requisitos. A rastreabilidade pós-requisitos refere-se aos aspectos da vida de um requisito que resultam da inclusão na especificação de requisitos. Rastreabilidade pré-requisitos é útil, por exemplo, quando há uma mudança de um requisito e é importante obter suas fontes ou as pessoas que o apoiam para validar a mudança. Rastreabilidade pós-requisitos é utilizada, por exemplo, para obter o módulo de projeto para o qual um requisito foi alocado, ou os procedimentos de teste criados para verificar o requisito. A Figura 2.2 apresenta as rastreabilidades pré-requisitos e pós-requisitos.

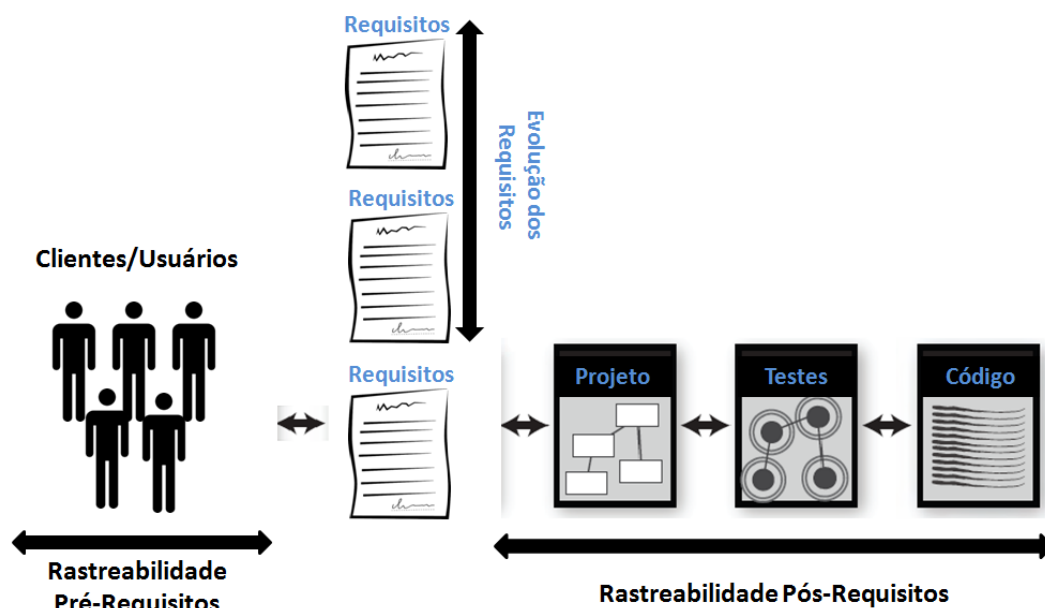


Figura 2.2 – Rastreabilidade Pré-Requisitos e Pós-Requisitos. Adaptado de (KANNENBERG; SAIEDIAN, 2009)

A capacidade de rastrear as ligações entre os requisitos é chamada de rastreabilidade inter-requisitos. As ligações entre os requisitos e outros artefatos são capturadas por rastreabilidade extra-requisitos. Rastreabilidade inter-requisitos é importante na análise de requisitos para lidar com mudanças e evolução de requisitos. Rastreabilidade extra-requisitos, por exemplo, é utilizada quando deseja-se saber todos os requisitos derivados de um determinado requisito, ou a sua cadeia de refinamento (PINHEIRO, 2004).

Embora a rastreabilidade de requisitos não possa ser completamente automatizada, porque o conhecimento das ligações é originado na mente dos membros da equipe de desenvolvimento, uma vez identificadas essas ligações, ferramentas de apoio são importantes para ajudar a gerenciar a grande quantidade de informações de rastreabilidade (WIEGERS, 2003). A utilização de ferramentas ajuda a aprimorar a precisão e reduz o tempo necessário para identificar relações de rastreabilidade (LUCIA *et al.*, 2007).

2.5 Linguagens de Modelagem de Requisitos

Uma linguagem de modelagem de requisitos tem como propósito a construção de modelos gráficos que simbolizam os artefatos dos componentes de software utilizados e os seus inter-relacionamentos com intuito de compreender melhor o sistema a ser desenvolvido (BOOCH; RUMBAUGH; JACOBSON, 2006).

Soares e Cioquetta (2012) destacam algumas das principais técnicas de modelagem de requisitos, incluindo entre outras, linguagem natural, linguagem natural estruturada, tabelas de decisão, histórias de usuário, diagrama de Caso de Uso da UML, diagrama de Requisitos da SysML e tabelas SysML. A técnica de linguagem natural é simples, mas tem como desvantagens as imprecisões, ambiguidades e inconsistências. A linguagem natural estruturada refere-se ao uso de especificações formatadas para escrever os requisitos de sistema tendo como principal característica conter a maior parte da facilidade de expressão e compreensão da linguagem natural, assegurando que algum grau de uniformidade seja imposto nas especificações. A linguagem natural estruturada é utilizada para dar mais estrutura nos documentos de requisitos. As tabelas de decisão fornecem uma notação que traduz ações e condições em um formato tabular. As histórias de usuário são usadas como parte da metodologia ágil eXtreme Programming (XP). Elas podem ser escritas pelo cliente usando terminologia não técnica no formato de frases usando linguagem natural. A história de usuário é uma declaração informal de um requisito de usuário em vez de um grande documento de requisitos. Embora a XP ofereça algumas vantagens no processo de ER, como o envolvimento do usuário e os formatos definidos para os requisitos e tarefas dos usuários, os requisitos ainda estão vagamente relacionados, não especificados graficamente e orientados para uma metodologia específica. O diagrama de Caso de Uso da UML apresenta os requisitos funcionais com simplicidade atuando como elo de comunicação entre as equipes técnicas e de negócio. O diagrama de Requisitos da SysML possui a flexibilidade de representar diversos tipos de requisitos (inclusive requisitos não funcionais) além de mostrar os vários tipos de relacionamentos entre diferentes requisitos. Nas tabelas SysML são representados os requisitos, suas propriedades e seus relacionamentos em forma tabular. Nesta seção, será detalhado o diagrama de Caso de Uso da UML e na Seção 2.6 o diagrama de Requisitos e tabelas da SysML.

O diagrama de Caso de Uso mostra as funcionalidades do sistema que são realizadas

através da interação do sistema com seus atores. A ideia é representar o que o sistema executará, não como. Os diagramas são compostos por atores, casos de uso e seus relacionamentos. Os atores podem corresponder a usuários, outros sistemas ou qualquer entidade externa ao sistema. A sequência detalhada de eventos em um caso de uso pode ser representada de maneiras diferentes. É comum descrever a sequência de eventos em linguagem natural estruturada com base em um padrão pré-definido ou por meio de diagramas de Atividades ou diagramas de Sequência. Uma limitação importante dos diagramas de Caso de Uso é a restrição de especificar apenas requisitos funcionais. Requisitos não funcionais, como desempenho, segurança, confiabilidade e requisitos externos, como interfaces, não são bem representados por diagramas de Casos de Uso (SOARES; VRANCKEN; VERBRAECK, 2011). A Figura 2.3 apresenta um diagrama de Casos de Uso.

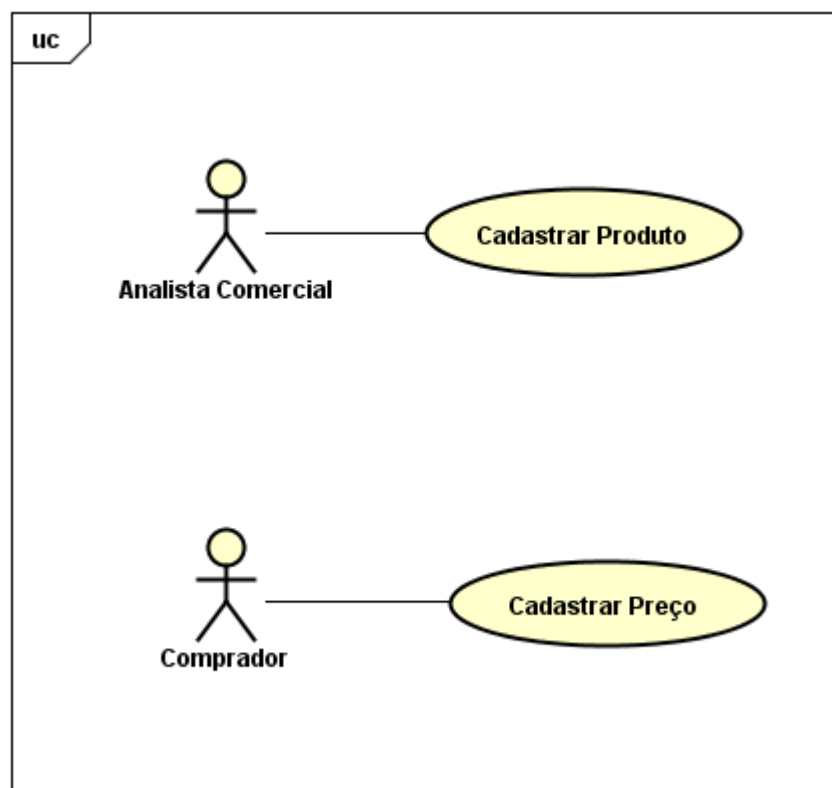


Figura 2.3 – Diagrama de Caso de Uso em UML

A SysML oferece recursos adicionais à UML, incluindo modelagem de requisitos (MARQUES; SIEGERT; BRISOLARA, 2014). A SysML oferece o diagrama de Requisitos que ajuda a organizar melhor os requisitos mostrando explicitamente os vários tipos de relações entre os diferentes requisitos. Outra vantagem de usar este diagrama é padronizar a maneira de especificar os requisitos definidos por meio de uma semântica (SOARES; VRANCKEN, 2008). Como consequência, SysML permite a representação de requisitos como elementos do modelo, tornando os requisitos parte da arquitetura do sistema (BALMELLI *et al.*, 2006).

Schneider, Naughton e Berenbach (2012) mencionam que a UML e a SysML são centradas na fase de projeto e que ambas se concentram na descrição técnica do sistema visualizado.

Eles propõem uma linguagem de modelagem centrada nos requisitos que capturam as informações essenciais da ER utilizada nas fases iniciais do projeto. A criação de modelos nesta linguagem permite ao engenheiro líder negociar com o cliente e distribuir eficientemente essa informação aos especialistas que precisam saber sobre ela e informá-los sobre o contexto durante as primeiras iterações do projeto.

2.6 *Systems Modeling Language (SysML)*

A SysML é uma linguagem de modelagem que tem foco na Engenharia de Sistemas. De acordo com o *Object Management Group* (OMG), a SysML permite a especificação, análise, projeto, verificação e validação de sistemas complexos. Como uma evolução da UML 2.0, a SysML pode ser aplicada a modelos de sistemas que podem incluir software, hardware, informações, processos e pessoas. A linguagem é útil na especificação de requisitos, no comportamento do sistema, na estrutura do sistema, em alocações de elementos para modelos e nas restrições nas propriedades do sistema (SOARES; VRANCKEN, 2008).

Para a criação da SysML, alguns diagramas foram derivados da UML sem alterações significativas (Caso de Uso, Máquina de Estados, Sequência e Pacote), alguns são derivados com alterações (Atividade, Definição de Bloco, Bloco Interno) e há dois novos diagramas (Requisitos e Paramétricos). SysML tem sido muitas vezes aplicada em uma variedade de atividades importantes durante o ciclo de vida do software, incluindo a comunicação com os *stakeholders*, melhorando o conhecimento do sistema, a execução do modelo e a documentação para a manutenção (BRIAND *et al.*, 2014), (RIBEIRO *et al.*, 2017).

2.6.1 Diagrama de Requisitos

O diagrama de Requisitos da SysML tem como objetivos organizar os requisitos e mostrar os vários tipos de relacionamentos entre diferentes requisitos. Outra vantagem de usar este diagrama é padronizar a maneira de especificar requisitos por meio de uma semântica definida. A SysML permite a representação de requisitos como elementos do modelo, o que significa que os requisitos fazem parte da arquitetura do sistema (BALMELLI *et al.*, 2006). As construções de requisitos SysML destinam-se a fornecer uma ponte entre as ferramentas de gerenciamento de requisitos tradicionais e os outros modelos SysML.

A Figura 2.4 apresenta um requisito do diagrama de Requisitos da SysML. Um requisito é composto por nome, “Id” e “text”. A propriedade “Id” representa um código que identifica o requisito no documento de requisitos e a propriedade “text” apresenta um texto breve sobre o requisito. Outras propriedades podem ser adicionadas ao requisito, pois o diagrama de Requisitos permite extensões.

O diagrama de Requisitos da SysML permite várias maneiras de representar relacionamentos de requisitos. As relações podem melhorar a especificação de sistemas, pois elas podem

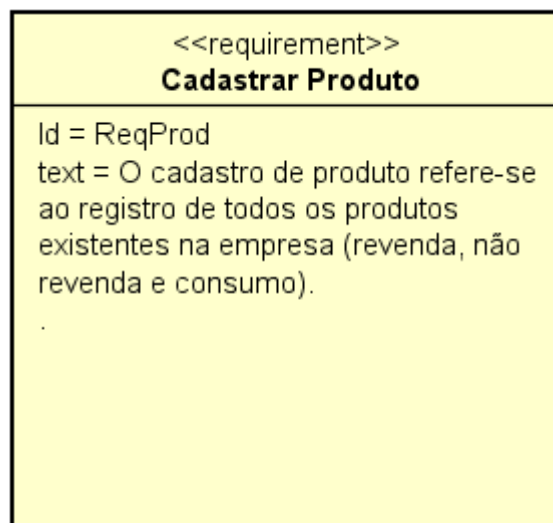


Figura 2.4 – Representação de um requisito - Diagrama de Requisitos em SysML

ser usadas para modelar requisitos (SOARES; VRANCKEN, 2008). As relações de derivação, refinamento, hierarquia, verificação, satisfação e rastreabilidade são brevemente explicadas a seguir.

A relação derivação relaciona uma exigência derivada com sua exigência de origem. Durante o processo de ER, novos requisitos são criados a partir dos anteriores. Normalmente, o requisito derivado está sob um requisito fonte na hierarquia.

A relação de refinamento descreve como um elemento de modelo (ou conjunto de elementos) pode ser usado para refinar posteriormente um requisito (SOARES; VRANCKEN, 2008).

A Figura 2.5 mostra um exemplo do diagrama de Requisitos da SysML de um sistema comercial com as relações de derivação (*deriveReq*) e refinamento (*refine*). A relação “*deriveReq*” relaciona o requisito “Cadastrar Preço” com seu requisito de origem (“Cadastrar Produto”) e o requisito “Cadastrar Produto” com seu requisito de origem (“Cadastrar Setor”). Por fim, a relação “*refine*” descreve com mais detalhes os requisitos “Cadastrar Seção”, “Cadastrar Grupo” e “Cadastrar Subgrupo”.

A relação de hierarquia permite a reutilização de requisitos. Neste caso, um requisito comum pode ser compartilhado por outros requisitos.

A relação verificação define como um caso de teste pode verificar um requisito. Isso inclui métodos de verificação padrão para inspeção, análise, demonstração ou teste.

A relação de satisfação descreve como um modelo satisfaz um ou mais requisitos. Esta relação representa a dependência entre um requisito e um elemento de modelo, como outros diagramas SysML, que representa esse requisito.

A relação de rastreabilidade fornece uma relação de propósito geral entre um requisito

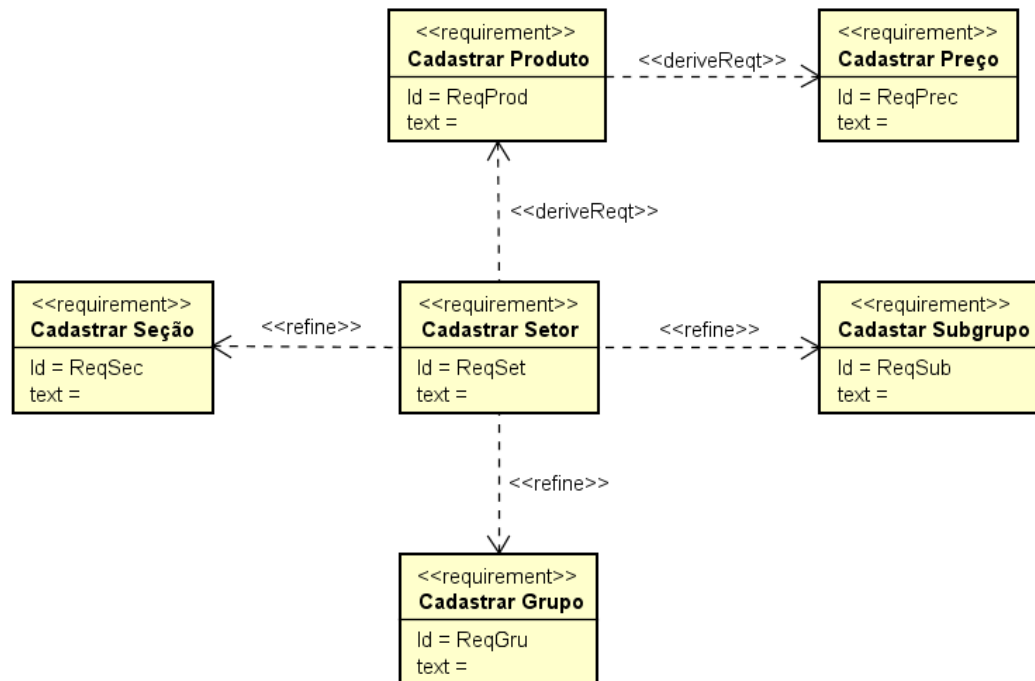


Figura 2.5 – Diagrama de Requisitos em SysML de um Sistema Comercial

e qualquer outro elemento do modelo. Sua semântica não tem restrições reais e não está bem definida como as outras relações. Por exemplo, uma dependência genérica de rastreamento pode ser usada para enfatizar que um par de requisitos estão relacionados de uma maneira diferente não definida por outros relacionamentos SysML (BALMELLI, 2007).

A Figura 2.6 mostra um exemplo do diagrama de Requisitos da SysML de um sistema de segurança com as relações de hierarquia, verificação (*verify*), satisfação (*satisfy*), rastreabilidade (*trace*) e derivação (*deriveReq*). A relação de hierarquia (representada pelo símbolo \oplus) é mostrada através do requisito “Ambiente Operacional” e seus requisitos “escravos” (“Operações Metereológicas” e “Operação 24/7”). A relação “*verify*” é utilizada para descrever o caso de teste “Teste de Pulverização de Água” usado para verificar o requisito “Operações Metereológicas”. A relação “*satisfy*” é utilizada para descrever que um elemento do modelo (Bloco “Câmera”) satisfaz o requisito “Decisão do Sensor”. A relação “*trace*” é utilizada para relacionar o requisito “Ambiente Operacional” a um elemento do modelo que representa uma fonte para o requisito (documento “Pesquisa de Mercado”). A relação “*deriveReq*” relaciona o requisito “Operações Metereológicas” com seu requisito de origem “Decisão do Sensor” e o requisito “Operação 24/7” com seu requisito de origem “Decisão do Sensor”.

2.6.2 Tabelas SysML

A linguagem SysML permite a representação de requisitos, suas propriedades e relacionamentos em um formato tabular. Uma tabela mostra a árvore hierárquica dos requisitos. Os campos propostos são o código do requisito, nome do requisito e o tipo de requisito (funcional

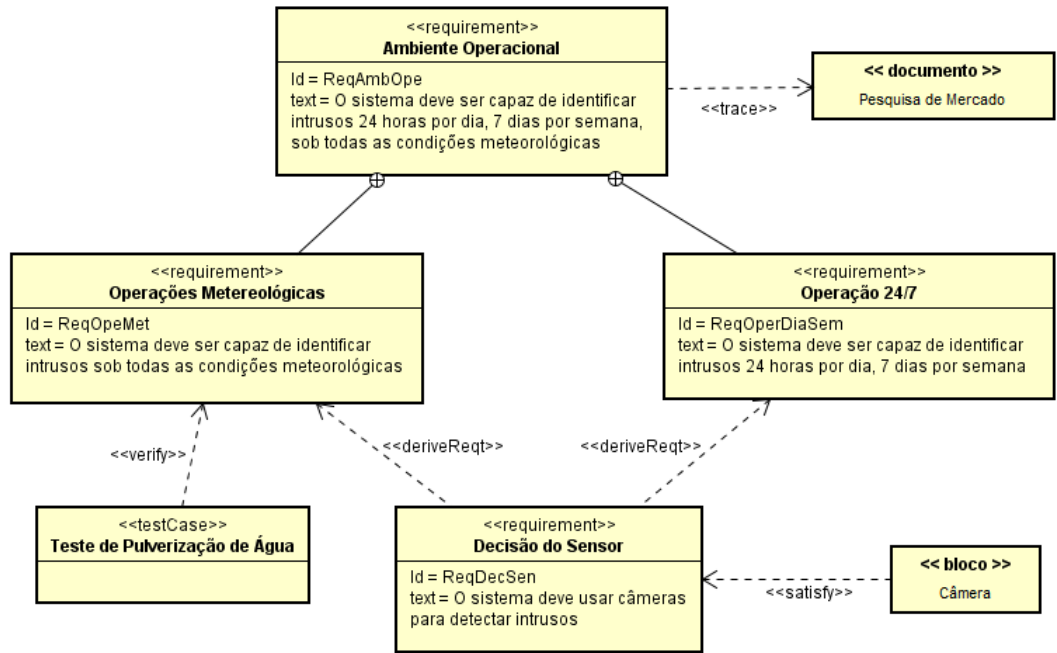


Figura 2.6 – Diagrama de Requisitos em SysML de um Sistema de Segurança. Adaptado de (FRIEDENTHAL; MOORE; STEINER, 2014)

ou não-funcional) e são apresentados na Tabela 2.1.

Tabela 2.1 – Uma tabela de requisitos em SysML (SOARES; VRANCKEN, 2008)

Código	Nome	Tipo

Outros itens de informação podem ser representados, como mostrado na Tabela 2.2. Por exemplo, o código do requisito, o nome do requisito, a qual requisito está relacionado (se houver), o tipo de relação e o tipo de requisito (funcional ou não-funcional). O tipo de relação refere-se aos tipos mencionados na Seção 2.6.1. De fato, sempre que um requisito é alterado ou excluído, as tabelas são úteis para mostrar que isso pode afetar diretamente outros requisitos.

Tabela 2.2 – Uma tabela de relações entre requisitos em SysML (SOARES; VRANCKEN, 2008)

Código	Nome	Requisitos relacionados	Tipo de relação (SysML)	Tipo

3 Técnicas e Ferramentas de Rastreabilidade de Requisitos

Este capítulo é baseado no artigo “*A Multi-Criteria Analysis of Techniques and Tools for Tracing Software Requirements*” publicado na Revista IEEE América Latina (JESUS, T. O. , SOARES, M. S.. A Multi-Criteria Analysis of Techniques and Tools for Tracing Software Requirements. IEEE Latin America Transactions, v. 15, p. 922-927, n. 2017).

O ciclo de vida da rastreabilidade de requisitos inclui quatro tarefas (EGYED *et al.*, 2009): aquisição, utilização, manutenção e aprimoramento. A aquisição compreende a criação de *links* de rastreabilidade entre requisitos e outros artefatos de software como código fonte e casos de teste. Estes *links* são criados manualmente ou com o auxílio de ferramentas. A utilização da rastreabilidade auxilia na análise de impacto de mudanças e na análise de dependência dos requisitos. A manutenção consiste na revisão contínua e na atualização dos *links* de rastreabilidade. Por fim, o aprimoramento melhora a qualidade dos *links* de rastreabilidade.

Regan *et al.* (2012) mencionam os principais problemas e barreiras para a implementação da rastreabilidade classificando-os em três categorias e apontando soluções para cada barreira identificada. As categorias foram divididas em problemas de gerenciamento, problemas sociais e problemas técnicos. Custos associados à implementação da rastreabilidade, poucas métricas para calcular o retorno sobre o investimento e falta de orientação dos profissionais para realização da rastreabilidade são exemplos dos problemas de gerenciamento. Com relação aos problemas sociais são citadas as barreiras das diferentes visões dos *stakeholders*, políticas internas e falta de comunicação entre as equipes. Nos problemas técnicos são mencionadas as dificuldades com as ferramentas, armazenamento e controle de versão das relações de rastreabilidade e complexidade da rastreabilidade (por exemplo, do grande número de artefatos e de rastreabilidade de requisitos não funcionais). Entre algumas soluções propostas para as barreiras, pode-se citar gestão da educação, criação de políticas organizacionais, integração de tarefas de rastreabilidade nas práticas de trabalho, criação de um repositório de artefatos integrado e utilização de técnicas e ferramentas de rastreabilidade.

Os métodos de rastreabilidade manuais, como matriz de rastreabilidade, envolvem a captura de dados de rastreabilidade de forma manual, o que para grandes projetos de software requer uma quantidade considerável de tempo e esforço (KANNENBERG; SAIEDIAN, 2009). A utilização de ferramentas ajuda a aprimorar a precisão e reduz o tempo necessário para identificar relações de rastreabilidade (LUCIA *et al.*, 2007) obtendo benefícios econômicos, técnicos e sociais (CYSNEIROS FILHO, 2011). Desta forma, é importante compreender as técnicas e as ferramentas de rastreabilidade de requisitos de software.

Neste capítulo, o objetivo é apresentar um estudo de técnicas e uma avaliação de ferramentas de rastreabilidade de requisitos de software a fim de avaliar se as técnicas são utilizadas na prática, e se estão sendo apoiadas por ferramentas. Um conjunto de critérios é proposto na Seção 3.4 para a avaliação das ferramentas.

3.1 Técnicas de Rastreabilidade de Requisitos

Torkar *et al.* (2012) realizaram uma revisão sistemática sobre rastreabilidade de requisitos. Um dos objetivos desta revisão foi identificar técnicas de rastreabilidade de requisitos. Neste trabalho, são brevemente explicadas as técnicas mais referenciadas no trabalho de Torkar *et al.* (2012), que são a rastreabilidade baseada em evento, recuperação da informação e rastreabilidade baseada em regras. Além destas técnicas, são apresentadas também a técnica mais utilizada na indústria (matriz de rastreabilidade) e uma técnica que envolve diretamente os *stakeholders* com um estudo de caso prático (rastreabilidade baseada em valor).

Rastreabilidade baseada em evento: A principal razão para o desenvolvimento desta técnica é fornecer melhor manutenção dos relacionamentos de rastreabilidade (TORKAR *et al.*, 2012). De acordo com Cleland-Huang, Chang e Christensen (2003), os relacionamentos de rastreabilidade são como relação *publish-subscribe* (EUGSTER *et al.*, 2003). Nesta relação, objetos dependentes, ou seja, artefatos, tem que se inscrever para os seus respectivos requisitos dos quais eles são dependentes. Sempre que ocorre uma mudança no requisito, uma mensagem de evento é publicada e todos os objetos dependentes são notificados (TORKAR *et al.*, 2012).

Na rastreabilidade baseada em evento, os objetos usam um servidor de eventos para se inscrever nos requisitos nos quais eles são dependentes e quando um evento de mudança significativo envolve um desses requisitos, uma mensagem de notificação de evento é publicada. A mensagem é recebida pelo servidor de eventos e reencaminhada para todos os objetos dependentes. Um gerenciador de assinantes lida com todas as notificações de eventos recebidas para um conjunto de objetos digitados de forma semelhante e segue protocolos predefinidos para lidar com tipos específicos de mensagens de eventos para esse objeto. Em alguns casos, o gerenciador de eventos pode atualizar automaticamente o objeto gerenciado, enquanto em outros casos a mensagem do evento deve ser armazenada em um *log* de eventos (CLELAND-HUANG *et al.*, 2002).

Recuperação da informação: É utilizada para automatizar a geração de *links* de rastreabilidade. Geralmente, utilizam métodos que incluem modelo de espaço vetorial, diferentes modelos probabilísticos e indexação semântica latente (MARCUS; MALETIC; SERGEYEV, 2005). A técnica consiste em três passos: (1) pré-processamento, (2) análise e indexação de documentos de entrada gerando uma representação para cada documento e (3) análise de artefatos utilizando algoritmos de classificação (ROCHIMAH; WAN-KADIR; ABDULLAH, 2007). Esta técnica reduz significativamente o esforço necessário para a criação dos *links* de rastreabilidade

entre os artefatos, mas exige esforços significativos para análise (TORKAR *et al.*, 2012).

Borg, Runeson e Brodén (2012) destacam que as ferramentas que implementam a recuperação da informação são baseadas em semelhanças textuais, por exemplo, entre requisitos e casos de teste. Borg, Runeson e Brodén (2012) propõem uma taxonomia de avaliações da técnica de recuperação da informação considerando quatro níveis de contexto: projeto, tarefas, busca e recuperação.

Rastreabilidade baseada em regras: A finalidade básica da técnica é gerar automaticamente *links* de rastreabilidade usando regras (SPANOUKAKIS *et al.*, 2004). Estas regras utilizam três documentos: especificação de requisitos, casos de uso e o diagrama de objetos (TORKAR *et al.*, 2012). Esta técnica consiste em quatro etapas: marcação gramatical dos artefatos, conversão dos artefatos em representação XML (*Extensible Markup Language*), geração das relações de rastreabilidade entre os artefatos e a geração de relações de rastreabilidade entre as diferentes partes dos artefatos (TORKAR *et al.*, 2012).

Mäder, Gotel e Philippow (2008) propuseram uma abordagem que se concentra exclusivamente na manutenção de relações de rastreabilidade durante a evolução e refinamento dos modelos UML estruturais utilizando a rastreabilidade baseada em regras. A criação inicial das relações de rastreabilidade entre esses artefatos pode ser estabelecida manualmente ou automaticamente. A abordagem foca no monitoramento das mudanças elementares que ocorrem nos elementos do modelo UML dentro de uma ferramenta CASE (*Computer-Aided Software Engineering*) e a geração de eventos de mudança baseados nessas, usando um conjunto de regras para ajudar a reconhecer esses eventos como partes constituintes de atividades de desenvolvimento. Uma vez identificadas essas atividades, as relações de rastreabilidade relacionadas aos elementos do modelo em mudança são atualizadas automaticamente.

Matriz de rastreabilidade: É uma tabela com o intuito de facilitar a visualização dos relacionamentos entre requisitos. A matriz de rastreabilidade é frequentemente utilizada na indústria de desenvolvimento de software para definir relações entre requisitos e outros elementos do software, por exemplo, projeto, código fonte e casos de teste. As matrizes são simples de criar e utilizar quando mantidas a um tamanho administrável (CLELAND-HUANG *et al.*, 2014). Além disso, são intuitivas, facilitando a compreensão pelos *stakeholders* (CLELAND-HUANG *et al.*, 2014). Geralmente, as matrizes de rastreabilidade são criadas manualmente e apresentam deficiências de escalabilidade e problemas de manutenção (CLELAND-HUANG, 2005; CLELAND-HUANG *et al.*, 2014). Um exemplo de matriz de rastreabilidade pode ser visualizado na Tabela 3.1.

Pode-se observar, por exemplo, que o requisito 1 impacta diretamente o requisito 3 (unidirecional), assim como o requisito 2 impacta diretamente o requisito 1 e com o requisito 4 (análise horizontal dos requisitos). Nesse caso, se por algum motivo existir a necessidade de se alterar o requisito 2, os requisitos 1 e 4 poderão ser afetados. A existência das relações de rastreabilidade possibilita identificar as origens das funcionalidades apresentadas no sistema.

Tabela 3.1 – Matriz de rastreabilidade

	Requisito 1	Requisito 2	Requisito 3	Requisito 4
Requisito 1			X	
Requisito 2	X			X
Requisito 3		X		X
Requisito 4			X	

Rastreabilidade baseada em valor: Esta técnica depende dos parâmetros definidos como importantes pelos *stakeholders*, como risco e volatilidade de um requisito (TORKAR *et al.*, 2012). O objetivo é identificar e rastrear os requisitos priorizados (TORKAR *et al.*, 2012). Esta técnica consiste em cinco etapas: definição de requisitos, priorização de requisitos, ligação dos requisitos com outros artefatos de software, agrupamento dos requisitos e uma avaliação final realizada pelo gerente do projeto (HEINDL; BIFFL, 2005). A técnica reduz o esforço de rastreabilidade apenas para os requisitos priorizados (HEINDL; BIFFL, 2005).

A Figura 3.1 mostra as etapas da técnica baseada em valor. Durante a fase definição de requisitos, o gerente de projeto ou engenheiro de requisitos analisa a especificação de requisitos de software para colocar um identificador exclusivo para cada requisito. O resultado da etapa de definição de requisitos é um conjunto de requisitos e seus códigos. Na fase priorização de requisitos, os *stakeholders* avaliam os requisitos com base em três parâmetros (o valor, o risco e o esforço de cada requisito). O resultado desta fase é uma lista ordenada de requisitos prioritários com base nos três níveis de prioridade. A fase agrupamento dos requisitos é opcional e permite que os arquitetos de software identifiquem grupos de requisitos. Estes grupos ajudam a desenvolver e refinar a arquitetura de um determinado conjunto de requisitos. Durante a fase ligação de artefatos, a equipe identifica vínculos de rastreabilidade entre requisitos e outros artefatos de software. Requisitos importantes são rastreados com mais detalhes do que outros requisitos e podem ser identificados a partir da lista de requisitos prioritários com base em três níveis desenvolvidos na etapa de priorização de requisitos. Um plano de rastreabilidade é o subproduto desta fase. Finalmente, durante a fase de avaliação final, o gerente de projeto pode usar a rastreabilidade para vários fins, como estimar o impacto de uma mudança (TORKAR *et al.*, 2012).

Heindl e Biffel (2005) realizaram um estudo de caso real utilizando a técnica baseada em valor. A técnica é avaliada em um projeto real com o intuito de analisar os custos e os benefícios com relação a rastreabilidade *ad hoc* e a rastreabilidade de todos os requisitos. Como resultado, a técnica baseada em valor levou cerca de 35% do esforço em comparação com rastreabilidade de todos os requisitos e que os requisitos com mais riscos precisam ser rastreados com mais detalhe. A fim de otimizar o valor da técnica, percebeu-se a necessidade de melhoria na etapa da priorização dos requisitos. Atributos como relevância arquitetural e estabilidade são mais importantes quando comparados ao valor do requisito, o risco e o esforço que foram considerados no trabalho. Por fim, a técnica baseada em valor é uma abordagem promissora para

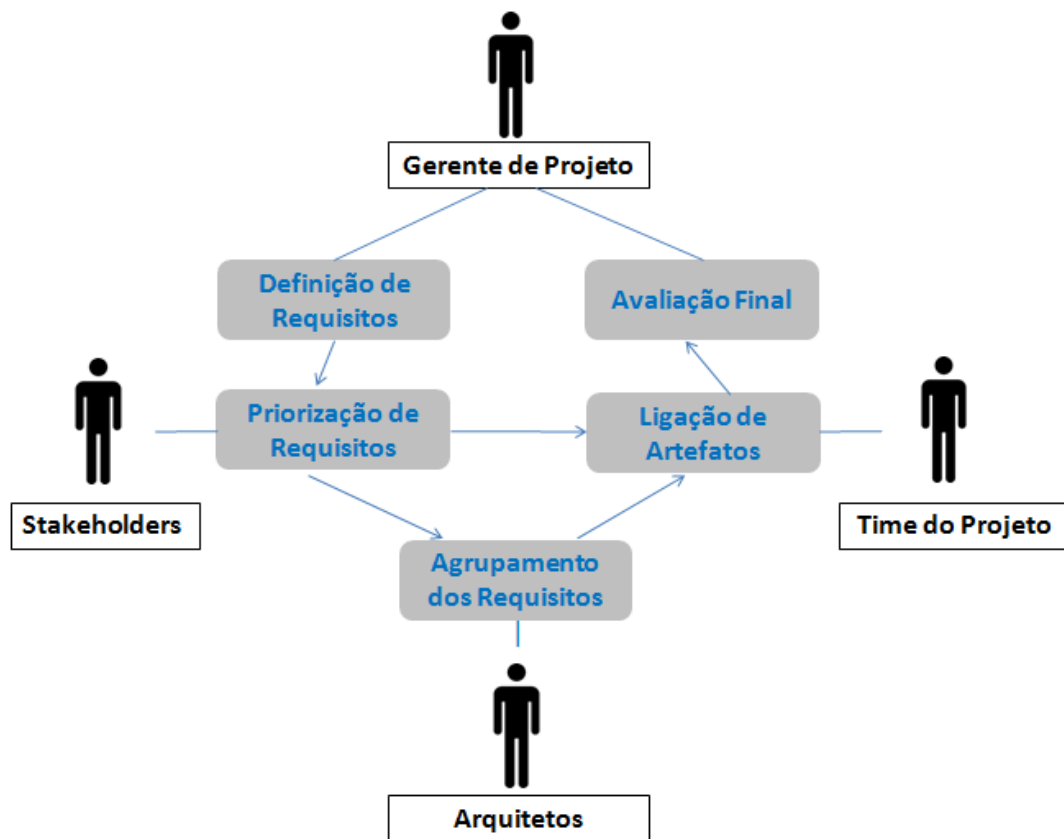


Figura 3.1 – Rastreabilidade baseada em valor. Adaptado de (TORKAR *et al.*, 2012)

diminuir o problema do elevado esforço da rastreabilidade de requisitos de uma forma prática e compreensível.

3.2 Avaliação das Técnicas Pesquisadas

Com relação às técnicas de rastreabilidade, a técnica baseada em evento é discutida em nove artigos do trabalho de Torkar *et al.* (2012), mas não há registro de evidências empíricas, ou seja, experimentos ou casos de estudo. Somente nas técnicas de recuperação da informação e na rastreabilidade baseada em valor foram observadas aplicações em estudos de casos. As técnicas citadas são aplicadas aos requisitos funcionais e não-funcionais, exceto a técnica de rastreabilidade baseada em valor que permite somente requisitos funcionais.

Todas as técnicas mencionadas na Seção 3.1 são relacionadas à rastreabilidade pós-requisitos, ou seja, refere-se aos aspectos da vida de um requisito que resultam após a inclusão na especificação de requisitos (WOHLIN; AURUM, 2005). Portanto, envolve a rastreabilidade entre os requisitos com artefatos gerados após a fase da especificação de requisitos como documento de projeto, código fonte e caso de teste.

Na Tabela 3.2, é apresentada uma análise comparativa entre as técnicas destacando as vantagens e desvantagens.

Tabela 3.2 – Comparativo entre técnicas de rastreabilidade de requisitos

Técnicas	Vantagens	Desvantagens
Baseada em evento	Manutenção dos <i>links</i> .	Excesso de mensagens (sistemas grandes).
Recuperação da informação	Automatização dos <i>links</i> . Redução de custos.	Esforço na análise.
Baseada em regra	Automatização dos <i>links</i> . Redução de custos.	Necessidade de documentos. Pouca flexibilidade.
Matriz de rastreabilidade	Simplicidade. Intuitiva. Fácil compreensão.	Escalabilidade. Problemas de manutenção.
Baseada em valor	Priorização dos requisitos. Redução de esforço.	Priorização sem critérios. Priorização subjetiva.

3.3 Ferramentas de Rastreabilidade de Requisitos

Informações de rastreabilidade podem ser capturadas manualmente por meio de técnicas como matrizes de rastreabilidade. As matrizes são tipicamente criadas usando uma planilha e mantidas de forma manual.

A utilização de uma ferramenta de apoio à gestão da rastreabilidade durante todo o ciclo de vida de desenvolvimento do software ajuda na precisão e na redução do tempo para identificar as relações de rastreabilidade (LUCIA *et al.*, 2007), (HONG; KIM; LEE, 2010), (CLELAND-HUANG; GOTEL; ZISMAN, 2012).

Neste trabalho, foram realizadas pesquisas na Internet buscando ferramentas de rastreabilidade de requisitos. Foram identificados os seguintes *sites*:

- www.volere.co.uk/tools.htm;
- www.scenarioplus.org.uk/vendors.htm;
- makingofsoftware.com/resources/list-of-rm-tools.

Estes *sites* mencionam as principais ferramentas de rastreabilidade de requisitos. Inicialmente, foram selecionadas doze ferramentas encontradas em pelo menos dois dos *sites* citados. As ferramentas são Caliber RM, Case SPEC, Coognition Cockpit, DevSpec, DOORS, In-Step RED, LiteRM, ReqSuite, Requisite Pro, ReqView, TopTeam e Yonix. Em seguida, foram realizados pré-cadastros nos *sites* das ferramentas para possíveis liberações e utilizações destas. Neste pré-cadastro, não houve êxito com as ferramentas Case SPEC, Coognition Cockpit, LiteRM, ReqSuite, TopTeam e Yonix, pois não houve retorno via *e-mail* do pré-cadastro. Posteriormente, foram realizados os *downloads* e as instalações das ferramentas que tiveram o pré-cadastro previamente aprovado. Dentre estas ferramentas, houve problema na utilização da ferramenta DevSpec por causa da necessidade de uma senha não enviada ao executar a ferramenta.

Desta forma, foram selecionadas para avaliação as ferramentas Caliber RM, DOORS, In-Step RED, Requisite Pro e ReqView. Por fim, estas ferramentas foram avaliadas utilizando um subconjunto de requisitos de um sistema de gerenciamento de trânsito (SOARES; VRANKEN, 2008). Esta lista de requisitos é um trabalho de uma auditoria técnica com base em um estudo de literatura e entrevistas. Os requisitos foram identificados por meio de entrevistas com *stakeholders*.

3.4 Critérios de Avaliação das Ferramentas

Para realizar o gerenciamento de requisitos de forma eficaz e eficiente, os membros da equipe de um projeto precisam utilizar uma ferramenta de apoio à gestão de requisitos durante todo o ciclo de desenvolvimento do software (HONG; KIM; LEE, 2010).

De acordo com Hong, Kim e Lee (2010), os critérios para avaliação das ferramentas de rastreabilidade de requisitos que auxiliam o gerenciamento da rastreabilidade são:

- **Armazenamento de atributos de requisitos:** Um projeto deve armazenar uma variedade de informações, ou atributos, sobre cada requisito. Os membros do projeto devem ser capazes de visualizar os atributos de cada requisito. Os responsáveis pelos requisitos devem ter a capacidade de modificar seus valores.

O responsável pelo requisito especifica a pessoa, grupo de *stakeholders*, organização ou unidade organizacional responsável pelo conteúdo de um requisito (POHL; RUPP, 2011).

O status de um requisito pode abranger diversos aspectos, como conteúdo, validação e de acordo. Status do conteúdo especifica o status atual do conteúdo do requisito (por exemplo: ideia, conceito, conteúdo detalhado). Status da validação especifica o status atual da validação (por exemplo: não validado, contém erro, em processo de correção). Status de acordo especifica o status atual da negociação de acordo (por exemplo: não acordado, acordado, em conflito) (POHL; RUPP, 2011).

A prioridade de um requisito especifica o nível de importância do requisito com relação às propriedades de priorização selecionadas (por exemplo: ordem de implementação, importância para aceitação pelo mercado, custo da oportunidade perdida em caso de não implementação) (POHL; RUPP, 2011). Neste trabalho foi considerada a prioridade com relação a ordem de implementação. A priorização é relevante porque os usuários têm expectativas em relação ao software em desenvolvimento. A entrega dos requisitos segundo a percepção dos usuários, além de contribuir para a satisfação deles, pode ser essencial para a continuidade do projeto (CORDEIRO; FREITAS, 2012).

Quando refere-se à priorização de requisitos, são utilizados os termos essencial, importante e desejável. A priorização essencial refere-se ao que realmente é fundamental para o software, sem o qual ele não pode ser dado como completo ou apto para a implantação.

São requisitos que se não são implementados impedem a implantação do software. A priorização importante deve ser parte do escopo, mas não bloqueia a implantação do software. Neste caso, o software contém uma pendência de escopo a ser atendido em momento oportuno. Este tipo de requisito pode ser postergado após a implantação, ou ser atendido temporariamente por soluções de contorno ou paliativos. Por fim, a priorização desejável refere-se ao requisito que não é indispensável para o software estar completo para a implantação. Também não é algo que, mesmo postergado, deverá ser feito obrigatoriamente. Sem um requisito desejável, o software deve funcionar de maneira satisfatória, atendendo completamente seus objetivos. Por ser algo que não precisa ser feito para que o software esteja completo, é considerada a menor das prioridades (VENTURA, 2013).

Também conhecido como esforço, a dificuldade de um requisito refere-se ao esforço estimado e/ou efetivo para implementar o requisito (POHL; RUPP, 2011). Este atributo é importante em fábricas de software, pois precisa receber uma especificação de requisitos de software explícita e completa porque a especificação corresponde ao produto final esperado.

A estabilidade de um requisito especifica a constância aproximada do requisito. A estabilidade é o número de mudanças que podem ser esperadas em relação ao requisito (POHL; RUPP, 2011). Por outro lado, a maturidade de um requisito refere-se à relação entre decisões sobre requisitos pendentes e que já foram tomadas. Com relação à estabilidade, os possíveis graus de estabilidade são classificados em fixo, estabilizado e volátil (POHL; RUPP, 2011).

- **Comunicação com os *stakeholders*:** O gerenciamento de requisitos envolve a comunicação entre os membros da equipe do projeto e os *stakeholders* e a adequação às mudanças de requisitos ao longo do curso do projeto. As ferramentas de rastreabilidade de requisitos devem permitir discutir questões do projeto de forma eletrônica. Como exemplo, mensagens eletrônicas devem ser notificadas aos *stakeholders* quando uma discussão é gerada ou um requisito é alterado (HONG; KIM; LEE, 2010).
- **Representação gráfica:** Para a visibilidade da rastreabilidade de requisitos, a ferramenta deve mostrar uma representação gráfica. Gráfico ou diagrama é uma forma mais eficaz para representar os tipos de rastreabilidade (HONG; KIM; LEE, 2010). A rastreabilidade em diagramas ajuda na análise do impacto das alterações mais claramente quando os componentes de desenvolvimento específicos são alterados (HONG; KIM; LEE, 2010).
- **Representação matricial:** É a forma mais comum de visualização da rastreabilidade de requisitos. A matriz de rastreabilidade pode ser definida como uma tabela que ilustra as ligações lógicas entre os requisitos funcionais individuais e outros artefatos do software. As matrizes são tipicamente criadas usando uma planilha e mantidas manualmente.

- **Agrupamento de requisitos:** Para facilitar o entendimento do conjunto de requisitos de um software, é conveniente interpretar que todo requisito pertence a algum agrupamento específico. O agrupamento de requisitos auxilia buscar os requisitos de forma mais rápida, além de estar associado a alguma funcionalidade ou tema específicos.

3.5 Avaliação das Ferramentas Pesquisadas

Com relação às ferramentas de rastreabilidade, na Tabela 3.3 é apresentado um guia para ajudar a selecionar qual ferramenta se adapta melhor ao software e às necessidades dos *stakeholders*. Os critérios foram classificados como totalmente atendidos (●) ou não atendidos (○) (ou não é facilmente atendido, ou mal atendido).

Tabela 3.3 – Resumo da avaliação de ferramentas de rastreabilidade de requisitos

Crítérios e Ferramentas	ReqView	In-Step RED	DOORS	Requisite Pro	Caliber RM
Responsável	○	●	○	●	●
Dificuldade	○	○	●	●	○
Prioridade	○	●	●	●	●
Estabilidade	○	●	○	●	○
Status	●	●	●	●	●
Comunicação c/ <i>Stakeholders</i>	○	○	○	○	○
Representação Gráfica	○	●	○	●	●
Representação em Matriz	●	○	○	●	●
Agrupamento de Requisitos	●	○	○	●	●

Requisite Pro e Caliber RM foram as ferramentas em que foram identificados a presença do maior número de critérios propostos deste trabalho. Nas ferramentas ReqView e DOORS foram identificados somente três critérios propostos entre os nove avaliados. Somente em duas das cinco ferramentas pesquisadas identificou-se a utilização dos critérios de dificuldade e estabilidade. Não foi identificada nenhuma ferramenta que utiliza o critério de comunicação com os *stakeholders* de forma automática.

Somente a ferramenta In-Step RED utiliza o diagrama de Requisitos da SysML. Este é um diagrama importante para a rastreabilidade de requisitos por auxiliar e mostrar de forma clara as relações entre requisitos de um software. Como a SysML é uma linguagem baseada na UML, a utilização deste diagrama para equipes que conhecem UML é realizada com menos esforço, considerando que UML é utilizada na academia e na indústria (SOARES; VRANCKEN, 2008).

Com relação à análise de técnicas e ferramentas em conjunto, as ferramentas geralmente não utilizam as técnicas de rastreabilidade pesquisadas. Somente a técnica de matriz de rastreabilidade foi identificada nas ferramentas ReqView e Caliber RM. Isto mostra que existe uma lacuna entre as técnicas e ferramentas de rastreabilidade de requisitos podendo ser abordadas em novos trabalhos e em novas soluções tecnológicas. Em muitos casos, as técnicas são criadas, mas não são homologadas em estudos de casos e experiências na indústria.

3.6 Conclusão do Capítulo

Neste capítulo, foi realizada uma revisão de técnicas de rastreabilidade de requisitos e uma avaliação de ferramentas a partir de um conjunto de critérios propostos. A fonte para esta avaliação foi uma pesquisa na Internet sobre as principais técnicas e ferramentas. Esta pesquisa pode ser vista como uma diretriz para a escolha de técnicas, critérios e ferramentas referente à rastreabilidade de requisitos no ciclo de desenvolvimento de software.

Com base no projeto e atributos relevantes definidos pelos *stakeholders*, algumas técnicas podem ser consideradas mais adequadas do que outras. As técnicas de rastreabilidade de requisitos se complementam e podem ser combinadas, ou seja, em um projeto pode ser aplicada mais de uma técnica aproveitando as vantagens de cada uma. A maioria das técnicas pesquisadas não foi validada empiricamente, tornando, principalmente para a indústria, a sua utilização e aplicação mais difícil.

Há poucos estudos comparativos sobre a relação entre técnicas e ferramentas de rastreabilidade de requisitos. Este trabalho limitou-se a analisar as ferramentas sob ponto de vista dos critérios propostos. Cada projeto e cada organização possuem critérios e prioridades distintos.

As ferramentas avaliadas não dispõem de um tratamento automático no processo de alteração dos atributos dos requisitos. Como exemplo, uma mensagem automática pode ser enviada aos *stakeholders* quando um atributo de um requisito é modificado.

Com relação aos problemas da rastreabilidade mencionados no trabalho de Regan *et al.* (2012), este capítulo identificou um dos problemas sociais que se refere à falta de comunicação entre as equipes e um dos problemas técnicos que se refere às dificuldades com as ferramentas de rastreabilidade.

É possível criar uma técnica apoiada por uma ferramenta que utilize a técnica de rastreabilidade baseada em evento contendo um tratamento automático com relação à comunicação com os *stakeholders*. Esta técnica permite a melhoria na comunicação quando houver mudanças nos requisitos. Nesta técnica pode-se utilizar as relações SysML apresentadas na Seção 2.6.1 e as tabelas SysML descritas na Seção 2.6.2. Na implementação da ferramenta, podem ser incluídos os critérios menos identificados nas ferramentas avaliadas, como dificuldade e estabilidade dos requisitos, conforme mencionado na Seção 3.5.

A técnica de rastreabilidade entre requisitos proposta e uma ferramenta que apoia a técnica são descritas no Capítulo 4.

4 Proposta de uma Técnica e Ferramenta de Rastreabilidade entre Requisitos

Este capítulo é baseado no artigo “*An Event-Based Technique to Trace Requirements Modeled with SysML*” publicado nas *Proceedings da 17th International Conference on Computational Science and Its Applications* (ICCSA, 2017).

A gestão de comunicações em um projeto de software envolve processos que asseguram a geração, coleta, distribuição, armazenamento, recuperação e destino final da informação do projeto de forma oportuna e adequada. Esses processos fornecem as conexões críticas entre pessoas e informações necessárias para uma comunicação bem-sucedida. Os gerentes de projeto podem gastar tempo excessivo em se comunicar com a equipe do projeto, com os *stakeholders* e o patrocinador do projeto. Todos os envolvidos no projeto devem entender como as comunicações afetam o projeto como um todo. Entre esses processos de gestão, um processo importante é relacionado com os envolvidos no gerenciamento de comunicações para atender aos requisitos de um projeto e resolver os problemas ocorridos durante o desenvolvimento do software.

Este capítulo apresenta uma técnica para melhorar a comunicação com os *stakeholders* durante o desenvolvimento de *software* utilizando a rastreabilidade baseada em evento por meio de notificações por *e-mail* aos *stakeholders*. Na técnica proposta também são incorporados os conceitos das tabelas e relações da SysML, conforme discutido na Seção 2.6. Para apoiar a técnica proposta, uma ferramenta denominada REQTRACE foi desenvolvida.

4.1 Proposta da Técnica de Rastreabilidade entre Requisitos

Na análise das ferramentas de rastreabilidade de requisitos apresentada na Seção 3.5, nenhuma ferramenta avaliada contém nas suas funcionalidades o critério de comunicação automática com *stakeholders*. Esta comunicação é importante, pois permite melhor comunicação nas diversas mudanças de requisitos que ocorre durante o desenvolvimento do software. Na técnica proposta, este critério foi o elemento principal.

Para a utilização da comunicação automática, foi usada a técnica baseada em evento mencionada na Seção 3.1. Esta técnica é baseada no padrão de projeto *event notifier* (GUPTA; HARTKOPF; RAMASWAMY, 2000) e oferece uma solução para algumas das dificuldades inerentes à manutenção da rastreabilidade, estabelecendo vínculos de rastreabilidade fracamente acoplados por meio do uso das relações *publish/subscribe* (EUGSTER *et al.*, 2003) entre objetos dependentes (HUANG, 2002). Este tipo de rastreabilidade é projetado para lidar com a mudança de forma robusta e, portanto, é favorável à mudança evolutiva de longo prazo.

Com relação aos *stakeholders*, a técnica proposta considerou dois tipos: o *stakeholder* principal, que é o responsável diretamente pelo requisito, ou seja, a pessoa responsável pela especificação do requisito, e os *stakeholders* envolvidos, que são as pessoas que possuem atividades impactadas por algum requisito, mas que não são responsáveis diretamente pelo requisito. Como exemplo, um gerente de recursos humanos pode ser o *stakeholder* principal pelo requisito “gerar folha de pagamento” de um sistema, mas o gerente financeiro e o gerente contábil podem ser *stakeholders* envolvidos, pois este requisito impacta diretamente nos processos e procedimentos de suas equipes.

Entre as diversas formas de notificar as mudanças nos requisitos aos *stakeholders*, foi definida para a técnica proposta a utilização de *e-mails*. Na maioria das empresas, o *e-mail* ainda é a principal forma de comunicação, por ser simples, funcional e uma maneira de oficializar as informações enviadas e recebidas. Na técnica proposta é importante que seja considerado o *e-mail* corporativo do funcionário, pois permite que as mensagens fiquem arquivadas, o que facilita em uma eventual busca de informações.

Com relação ao modelo de *e-mail* a ser enviado aos *stakeholders*, a técnica define três modelos que são detalhados na Tabela 4.1. Para cada tipo de operação, é definido o modelo de *e-mail* com mensagens específicas no assunto e no corpo do *e-mail* e a quais tipos de destinatários (*stakeholders*) a notificação será enviada. Quando ocorre uma alteração no requisito, a expressão “Stakeholder involved” é adicionada no final do assunto do *e-mail* somente para os *stakeholders* envolvidos.

Tabela 4.1 – Modelos de *e-mails* (notificações aos *stakeholders*)

Operação	Modelo de <i>e-mails</i>	<i>Stakeholders</i>
Inserção	<p><i>Nome do Sistema</i> - New requirement was created: Requirement NN. (assunto)</p> <p><i>Nome do Sistema</i> - New requirement was created: Requirement NN - <i>Descrição do Requisito</i>. Please, check the system. (corpo)</p>	Principal
Alteração	<p><i>Nome do Sistema</i> - Requirement NN has been updated. (assunto)</p> <p><i>Nome do Sistema</i> - Requirement NN has been updated. <i>Descrição do Requisito</i> (<i>Nome do atributo</i> has been modified). Please, check the system. (corpo)</p>	Principal e envolvidos
Exclusão	<p><i>Nome do Sistema</i> - Requirement NN was deleted. (assunto)</p> <p><i>Nome do Sistema</i> - Requirement NN was deleted. <i>Descrição do Requisito</i>. Please, check the system. (corpo)</p>	Principal

A técnica proposta combinou a técnica de rastreabilidade de requisitos baseada em evento, a linguagem de modelagem SysML e o critério de comunicação automática com os *stakeholders*. A Figura 4.1 mostra os principais elementos propostos na técnica. A técnica utiliza

os conceitos de tabelas e relações SysML. A SysML foi escolhida por conter tipos de relações entre os requisitos conforme apresentado na Seção 2.6.1 e por ter sido muitas vezes aplicada em atividades importantes durante o ciclo de vida do software, incluindo a comunicação com os *stakeholders* (BRIAND *et al.*, 2014) (RIBEIRO *et al.*, 2017). As tabelas SysML são utilizadas com o objetivo de organizar os requisitos mapeando-os em um formato tabular. As tabelas SysML são descritas na Seção 2.6.2 deste trabalho.

Na Figura 4.1 é mostrada também a ferramenta REQTRACE que apoia a técnica proposta. Na ferramenta, o gerente de projeto tem como responsabilidade a inserção e a manutenção dos requisitos na ferramenta e os *stakeholders* (principais e envolvidos) são notificados via *e-mail* das mudanças realizadas no cadastro de requisitos pelo gerente do projeto. O detalhamento da ferramenta é apresentado na Seção 4.2.

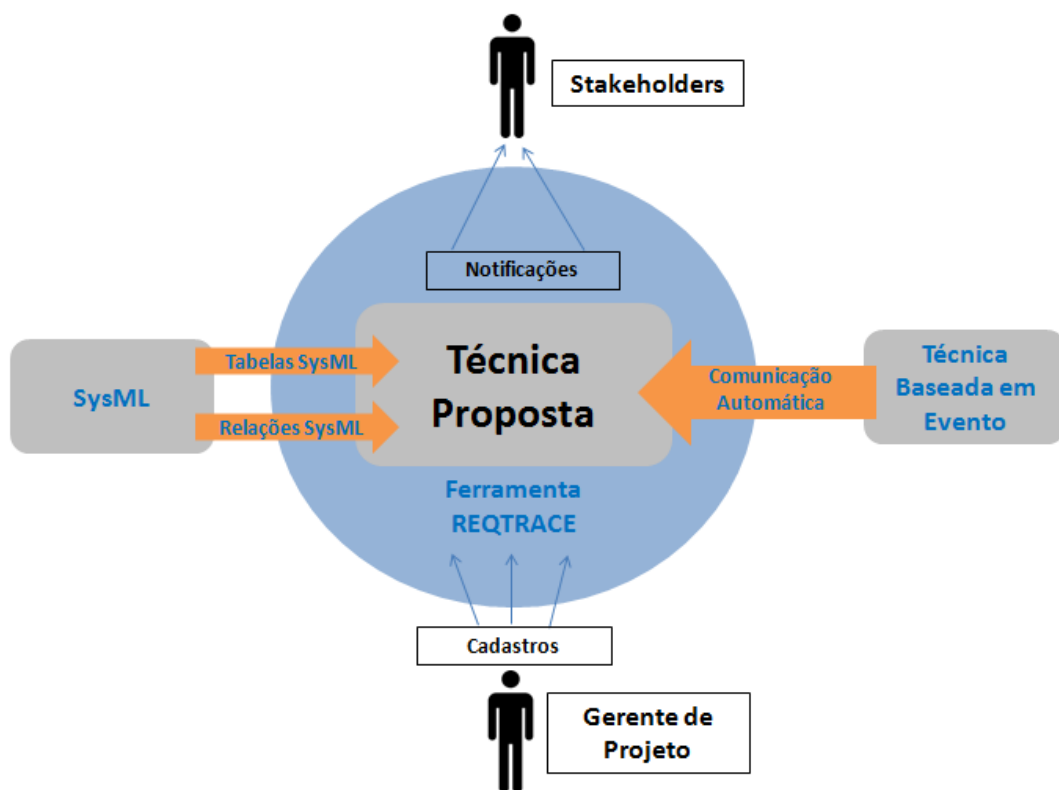


Figura 4.1 – Proposta da técnica de rastreabilidade entre requisitos

4.2 Funcionalidades da Ferramenta REQTRACE

A ferramenta REQTRACE foi desenvolvida usando a linguagem de programação C Sharp e o sistema de gerenciamento de banco de dados SQL Server. A escolha de C Sharp deve-se por esta linguagem proporcionar recursos de programação orientada a objetos, uso de eventos nos controles e validação de dados. A principal razão para a escolha do SQL Server foi devido ao suporte nativo à linguagem C Sharp.

Na implementação da ferramenta proposta, foi utilizada uma simplificação da técnica de rastreabilidade de requisitos baseada em evento, pois não foi implementado um servidor de eventos para assinar os requisitos dos quais eles são dependentes e, quando ocorre uma alteração significativa, como uma modificação ou atualização de requisitos envolvendo um desses requisitos, uma mensagem de notificação de evento é publicada. A disponibilização e configuração de um servidor de eventos demandaria complexidade e tempo para implementação da ferramenta. Com o intuito de reduzir a complexidade na implementação, utilizou-se o protocolo *Simple Mail Transfer Protocol* (SMTP) para o envio das notificações por *e-mail*. Esta implementação foi realizada na funcionalidade “cadastro de requisitos”.

Com relação ao envio dos *e-mails*, foi configurado um servidor SMTP que tem como responsabilidade o envio dos *e-mails* gerados pela ferramenta. O SMTP é o protocolo padrão para envio de *e-mails*. Foi utilizado o servidor GMail-Google por ter um servidor estável, além de assegurar desempenho adequado no envio das mensagens. A configuração do servidor SMTP foi realizada adicionando parâmetros diretamente na ferramenta REQTRACE (cadastro de requisitos). Para a conexão com o servidor, foi utilizado o protocolo SSL (*Secure Sockets Layer*) que permite flexibilidade e segurança no envio de *e-mails*.

A ferramenta envia o *e-mail* utilizando como remetente o *e-mail* “reqtraceebt@gmail.com” que foi criado somente para esta funcionalidade. Além disso, a ferramenta possui três cadastros (*stakeholders*, requisitos e relações entre requisitos) e duas consultas (requisitos e histórico de mudanças de requisitos). As Figuras 4.2 e 4.3 mostram as opções de cadastro e consulta da ferramenta REQTRACE.

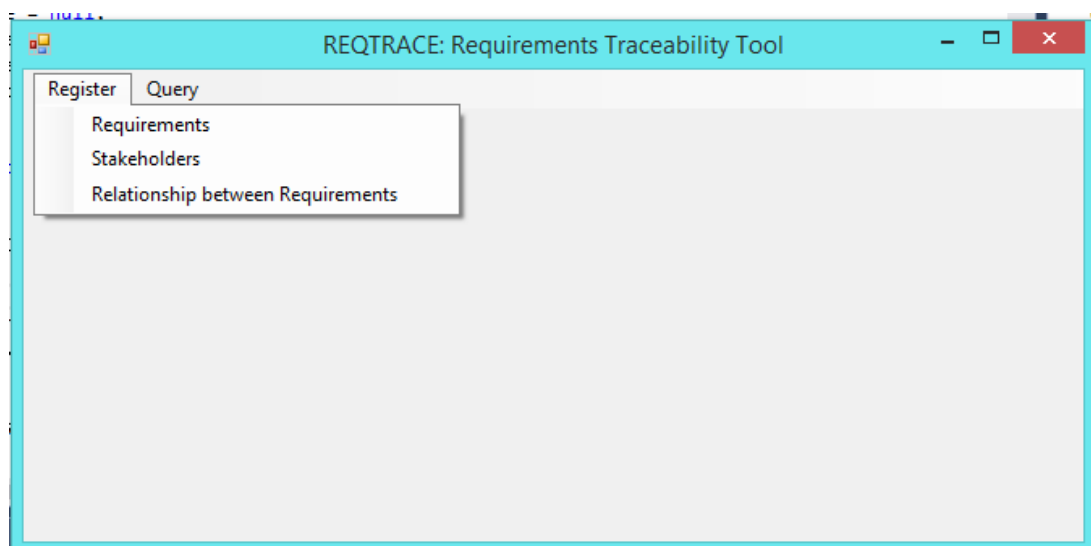
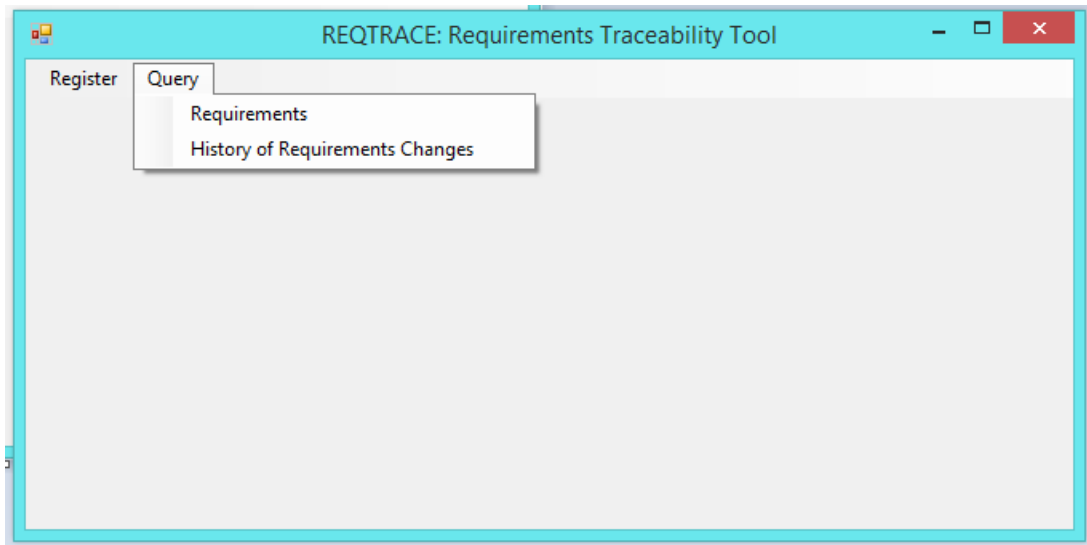


Figura 4.2 – Tela de opções do *menu* cadastro

Figura 4.3 – Tela de opções do *menu* consultas

O cadastro de *stakeholders* tem como objetivo armazenar todos os *stakeholders* envolvidos em qualquer sistema. O atributo *e-mail* é fundamental para a utilização e funcionamento da ferramenta.

O cadastro de requisitos destina-se a controlar todos os requisitos do sistema, com ênfase nas notificações por *e-mail* para os *stakeholders* (principal e envolvido(s)) do requisito. Cada requisito tem um *stakeholder* principal (obrigatório) com a possibilidade de registrar outros *stakeholders* envolvidos (opcional).

A relação entre requisitos tem como finalidade a conexão dos requisitos e seu tipo de relação mencionado na Seção 2.6. Esta é a funcionalidade da ferramenta que faz uso de tabelas de requisitos SysML. Um sistema é baseado em vários requisitos que podem ser relacionados uns aos outros.

A consulta de requisitos tem como objetivo visualizar todos os requisitos registrados no sistema.

O histórico de mudanças de requisitos rastreia todas as alterações realizadas no sistema. Todas as inclusões e modificações nos atributos são armazenadas para consultas adicionais. Este controle é baseado em gatilhos (*triggers*) ativados no sistema de gerenciamento de banco de dados.

O diagrama de Casos de Uso da UML da ferramenta é apresentado na Figura 4.4. O gerente de projeto é responsável pelos cadastros de *stakeholders*, requisitos e das relações entre requisitos. Os *stakeholders* têm acesso somente às consultas de requisitos e ao histórico de mudanças de requisitos.

O diagrama de Classes da UML da ferramenta REQTRACE é apresentado na Figura 4.5. A classe “*System*” está associada à classe “*Requirement*”, pois um sistema é constituído por

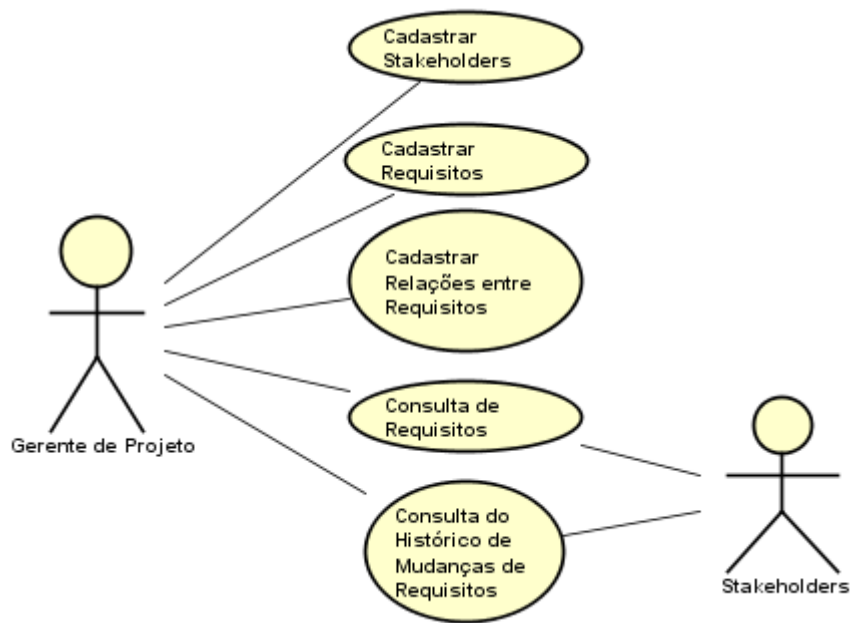


Figura 4.4 – Diagrama de Caso de Uso da UML da ferramenta REQTRACE

diversos requisitos. A classe “*Stakeholder*” está associada à classe “*Requirement*” em virtude do *stakeholder* ser responsável pelos requisitos. Um requisito possui status e contém mudanças nos seus atributos durante o desenvolvimento do software e são representadas no diagrama, respectivamente, pelas classes “*Status*” e “*Change*”. As relações SysML são representadas pela classe “*RelationshipSysML*” e as relações entre os requisitos pela classe “*RelationshipRequirement*”.

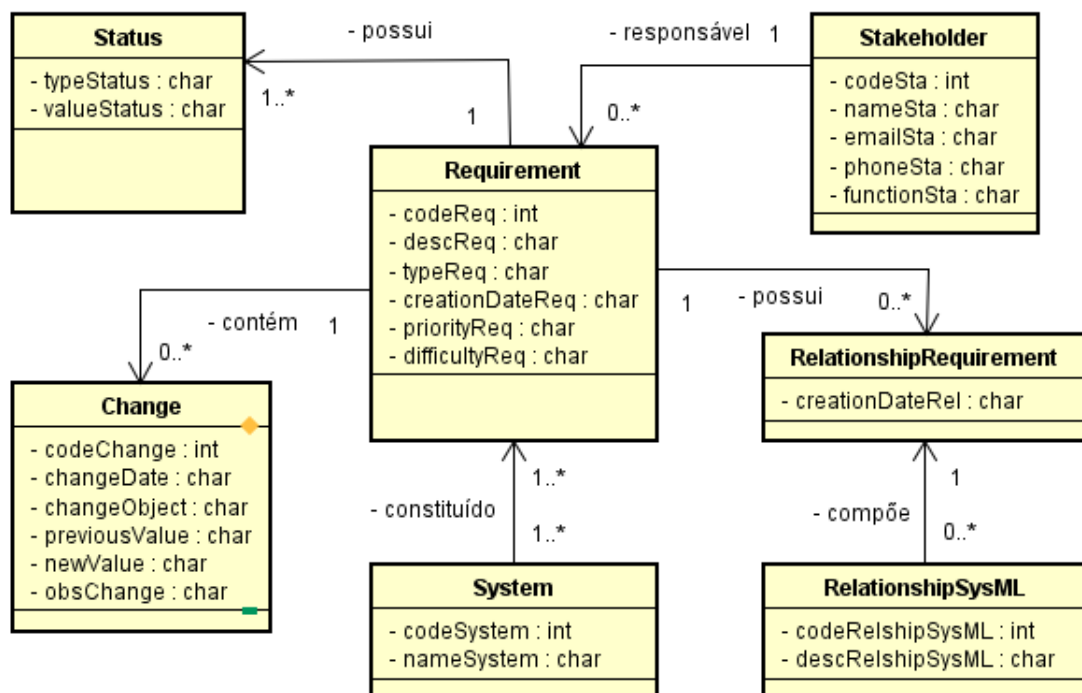


Figura 4.5 – Diagrama de Classes da UML da ferramenta REQTRACE

4.3 Telas da Ferramenta REQTRACE

Nesta seção são apresentadas as telas da ferramenta REQTRACE.

O cadastro de *stakeholders* tem como objetivo armazenar todos os *stakeholders* envolvidos em um dado sistema. Os atributos deste cadastro são: código do *stakeholder*, nome, *e-mail*, telefone e função. A Figura 4.6 mostra a tela referente ao cadastro.

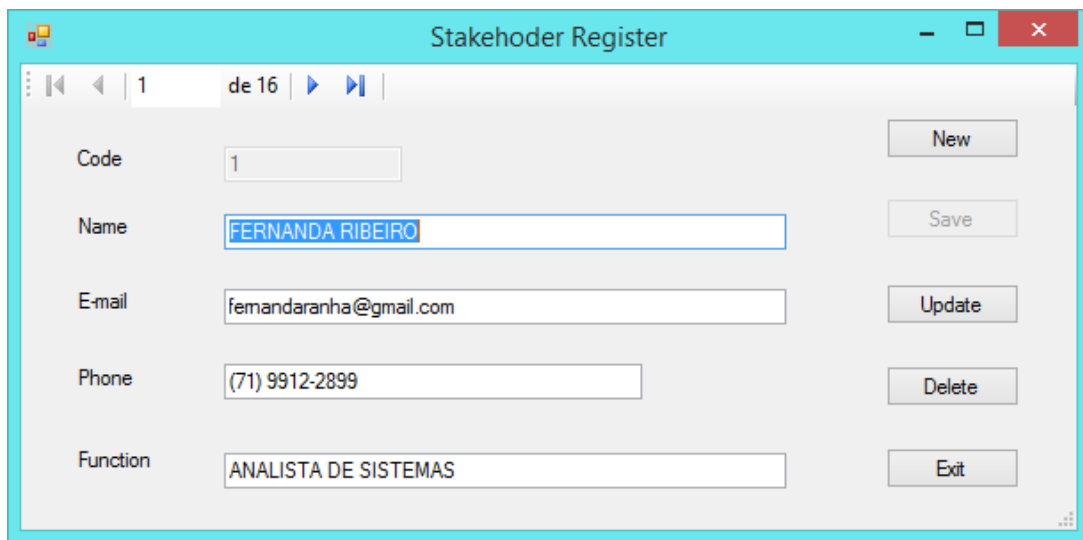


Figura 4.6 – Cadastro de *stakeholders*

O cadastro de requisitos destina-se a gerenciar todos os requisitos do sistema, com ênfase nas notificações por *e-mail* para os *stakeholders* (principal e envolvido(s)) do requisito.

Para cada requisito são armazenados os seguintes atributos: código do requisito, descrição do requisito, tipo do requisito (funcional, não-funcional, externo), data de criação (gerada pela ferramenta), prioridade (desejável, importante, essencial), esforço de implementação (baixo, médio, alto, muito alto), *stakeholder* principal, *stakeholder* envolvido(s), status (em correção, tem erro, não validado) e sistema ao qual o requisito pertence. A Figura 4.7 mostra a tela referente ao cadastro.

No cadastro de requisitos são realizadas as notificações por *e-mail* aos *stakeholders* quando os botões “Save”, “Update” e “Delete” são selecionados. A Figura 4.8 mostra o *e-mail* gerado na inclusão do requisito, a Figura 4.9 quando o requisito é modificado e a Figura 4.10 quando o requisito é excluído.

Figura 4.7 – Cadastro de requisitos

• Payment Sheet System - New requirement was created : Requirement 32 ★

reqtracebt@gmail.com Jun 19 em 9:13 PM
 Para jtelmooliveira@yahoo.com.br

Payment Sheet System - New requirement was created.
 Requirement 32 - Employee registration .
 Please, check the system.

Figura 4.8 – E-mail gerado pela ferramenta REQTRACE na inclusão de um requisito

• Payment Sheet System - Requirement 32 has been updated. Pessoas★

reqtracebt@gmail.com Jun 19 em 9:14 PM
 Para jtelmooliveira@yahoo.com.br

Payment Sheet System - Requirement 32 has been updated.
 Employee registration (Priority has been modified).
 Please, check the system.

Figura 4.9 – E-mail gerado pela ferramenta REQTRACE na modificação de um requisito

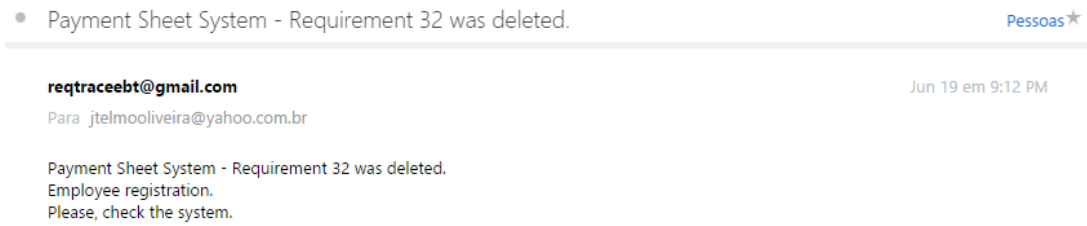


Figura 4.10 – E-mail gerado pela ferramenta REQTRACE na exclusão de um requisito

A relação entre requisitos tem como finalidade a associação entre os requisitos e seu tipo de relação mencionado na Seção 2.6. Esta é a funcionalidade da ferramenta que faz uso de tabelas de requisitos e das relações SysML. Um sistema pode ter várias relações entre os requisitos. Para cada relação cadastrada na ferramenta, são solicitados os dois requisitos que estão relacionados e o tipo da relação (hierarquia, satisfação, derivação, verificação, refinamento e rastreabilidade). Nesta funcionalidade, são mostradas duas tabelas. A primeira tabela (*Requirements*) apresenta todos os requisitos e na segunda tabela (*SysML Requirements Relationships*) são apresentadas as relações cadastradas entre os requisitos. A Figura 4.11 mostra a tela referente ao cadastro destas relações.

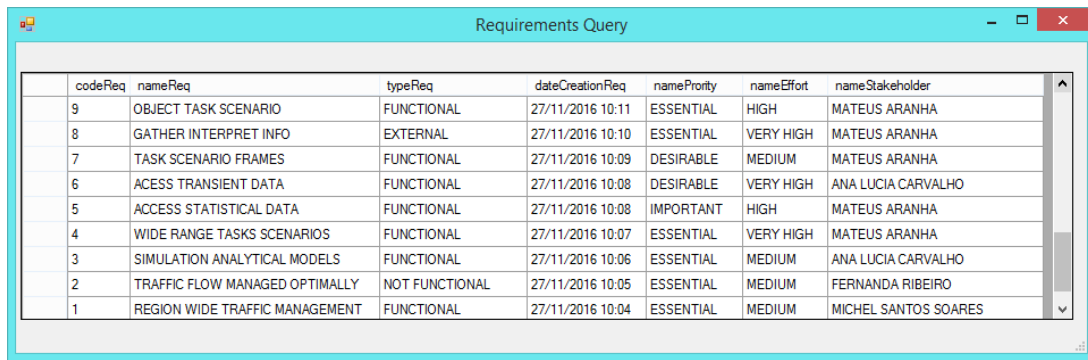
codeReq	nameReq	nameTypeReq
1	REGION WIDE TRAFFIC MANAGEMENT	FUNCTIONAL
2	TRAFFIC FLOW MANAGED OPTIMALLY	NOT FUNCTIONAL
3	SIMULATION ANALYTICAL MODELS	FUNCTIONAL
4	WIDE RANGE TASKS SCENARIOS	FUNCTIONAL
5	ACCESS STATISTICAL DATA	FUNCTIONAL
6	ACCESS TRANSIENT DATA	FUNCTIONAL
7	TASK SCENARIO FRAMES	FUNCTIONAL
8	GATHER INTERPRET INFO	EXTERNAL

codReq1	codReq2	nameReq	nameTypeReq	nameRelationship
7	1	REGION WIDE TRAFFIC MANAGEMENT	FUNCTIONAL	TRACE
7	2	TRAFFIC FLOW MANAGED OPTIMALLY	NOT FUNCTIONAL	HIERARCHY
8	3	SIMULATION ANALYTICAL MODELS	FUNCTIONAL	REFINE
9	4	WIDE RANGE TASKS SCENARIOS	FUNCTIONAL	DERIVE

Figura 4.11 – Cadastro das relações entre os requisitos

A consulta de requisitos tem como objetivo visualizar todos os requisitos registrados no sistema. A Figura 4.12 mostra a tela referente a esta consulta.

O histórico de mudanças de requisitos rastreia todas as alterações realizadas no sistema. Todas as inclusões e modificações nos atributos realizadas no cadastro dos requisitos são armazenadas para consultas adicionais. Nesta funcionalidade são mostradas as seguintes informações: código do requisito, tipo de mudança (inserção - INS, alteração - UPD, exclusão - DEL), data de

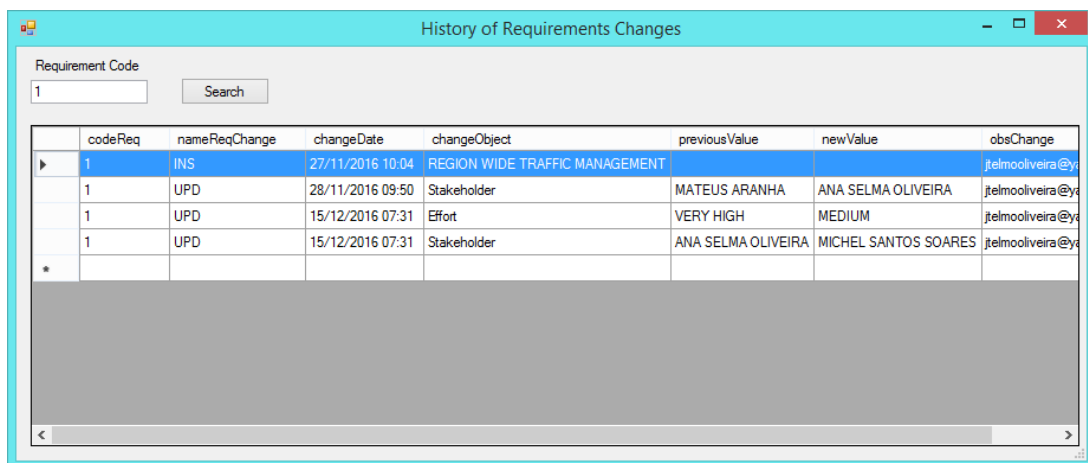


	codeReq	nameReq	typeReq	dateCreationReq	namePriority	nameEffort	nameStakeholder
9		OBJECT TASK SCENARIO	FUNCTIONAL	27/11/2016 10:11	ESSENTIAL	HIGH	MATEUS ARANHA
8		GATHER INTERPRET INFO	EXTERNAL	27/11/2016 10:10	ESSENTIAL	VERY HIGH	MATEUS ARANHA
7		TASK SCENARIO FRAMES	FUNCTIONAL	27/11/2016 10:09	DESIRABLE	MEDIUM	MATEUS ARANHA
6		ACCESS TRANSIENT DATA	FUNCTIONAL	27/11/2016 10:08	DESIRABLE	VERY HIGH	ANA LUCIA CARVALHO
5		ACCESS STATISTICAL DATA	FUNCTIONAL	27/11/2016 10:08	IMPORTANT	HIGH	MATEUS ARANHA
4		WIDE RANGE TASKS SCENARIOS	FUNCTIONAL	27/11/2016 10:07	ESSENTIAL	VERY HIGH	MATEUS ARANHA
3		SIMULATION ANALYTICAL MODELS	FUNCTIONAL	27/11/2016 10:06	ESSENTIAL	MEDIUM	ANA LUCIA CARVALHO
2		TRAFFIC FLOW MANAGED OPTIMALLY	NOT FUNCTIONAL	27/11/2016 10:05	ESSENTIAL	MEDIUM	FERNANDA RIBEIRO
1		REGION WIDE TRAFFIC MANAGEMENT	FUNCTIONAL	27/11/2016 10:04	ESSENTIAL	MEDIUM	MICHEL SANTOS SOARES

Figura 4.12 – Consulta de requisitos

modificação, atributo modificado, valor anterior à modificação, novo valor após a modificação, *e-mail* do *stakeholder* notificado no momento da modificação.

Esta consulta é importante, pois registra todas as mudanças que ocorrem nos atributos do cadastro de requisitos evitando questionamentos e divergências entre a equipe de desenvolvimento e *stakeholders*, que ocorrem durante o desenvolvimento do *software*. A Figura 4.13 mostra a tela referente a esta consulta.



	codeReq	nameReqChange	changeDate	changeObject	previousValue	newValue	obsChange
►	1	INS	27/11/2016 10:04	REGION WIDE TRAFFIC MANAGEMENT			jelmooliveira@y
	1	UPD	28/11/2016 09:50	Stakeholder	MATEUS ARANHA	ANA SELMA OLIVEIRA	jelmooliveira@y
	1	UPD	15/12/2016 07:31	Effort	VERY HIGH	MEDIUM	jelmooliveira@y
	1	UPD	15/12/2016 07:31	Stakeholder	ANA SELMA OLIVEIRA	MICHEL SANTOS SOARES	jelmooliveira@y
*							

Figura 4.13 – Consulta do histórico de mudanças nos requisitos

4.4 Conclusão do Capítulo

Uma das atividades relacionadas ao gerenciamento de requisitos é a rastreabilidade de requisitos, que é um mecanismo importante para gerenciar e auditar todo o processo de desenvolvimento de software. A rastreabilidade influencia a qualidade dos produtos de software, facilitando sua reutilização.

Este capítulo teve como objetivo mostrar uma técnica de rastreabilidade entre requisitos apoiada por uma ferramenta denominada REQTRACE.

A técnica proposta combina a técnica baseada em evento e conceitos da SysML (relações

e tabelas). Esta combinação permite melhor comunicação com os *stakeholders* sendo enviadas notificações por *e-mail* aos responsáveis e aos envolvidos pelos requisitos. O gerente de projeto tem como responsabilidade a manutenção do cadastro dos *stakeholders*, dos requisitos e das relações entre os requisitos.

Para apoiar a técnica, a ferramenta REQTRACE implementa os conceitos propostos da técnica. A ferramenta “envia” as notificações por *e-mail* aos *stakeholders* quando houver qualquer modificação nos atributos do cadastro de requisitos.

O próximo capítulo apresenta uma avaliação da técnica e da ferramenta propostas buscando avaliar se ambas contribuem para melhor monitoramento dos requisitos de um projeto de software. A avaliação foi realizada por 24 profissionais da área de tecnologia da informação. Para a avaliação, foram utilizadas as técnicas de questionário e entrevistas.

5 Avaliação da Técnica e da Ferramenta REQTRACE

Neste capítulo é apresentada uma avaliação da técnica e da ferramenta propostas na dissertação. As seções são apresentadas, nesta ordem: inicialmente, o modelo TAM é descrito na Seção 5.1, em seguida a avaliação quantitativa por meio de questionário na Seção 5.2, a avaliação qualitativa por meio de entrevistas na Seção 5.3 e, por fim, a conclusão do capítulo na Seção 5.4.

5.1 Modelo TAM

O *Technology Acceptance Model* (TAM) (DAVIS, 1989) é um modelo criado especificamente para o estudo do comportamento de uso de tecnologias de computador. O modelo TAM sugere que quando os usuários são apresentados a uma nova tecnologia, uma série de fatores influenciam sua decisão sobre como e quando a usarão. Originalmente, o TAM propõe duas variáveis principais: *Perceived Ease of Use* (PEOU) - facilidade de uso percebida e *Perceived Usefulness* (PU) - utilidade percebida. Uma terceira variável é considerada nesta pesquisa: *Perceived Usage* (PUE) - uso percebido, como introduzido em (ADAMS; NELSON; TODD, 1992).

PEOU é uma variável que pode ser definida como o grau em que uma pessoa acredita que o uso de um sistema de informação/tecnologia será livre de esforço (DAVIS, 1989). É importante saber se haverá alguma resistência dos usuários para a nova tecnologia. Esta variável refere-se às expectativas do indivíduo na isenção do esforço físico ou mental para o uso de um determinado sistema ou tecnologia. Software pode ser difícil de usar, frustrando os usuários, o que pode até levar a uma diminuição em seu desempenho para finalizar suas tarefas.

PU pode ser definido como o grau em que uma pessoa acredita que o uso de um sistema/ferramenta pode melhorar o desempenho de suas atividades (DAVIS, 1989). PU está relacionado como os usuários acham que o software irá ajudá-los a alcançar seus objetivos em suas tarefas no trabalho.

PUE é a variável que procura apontar as características que são realmente usadas no software. Esta variável localiza quais são os recursos que têm o uso mais importante na ferramenta. Dada a importância certa para os recursos, pode-se ver como o software será utilizado, quais recursos serão usados e aqueles não considerados pela maioria dos usuários.

A avaliação foi realizada por meio de duas técnicas: questionário, baseado no modelo TAM, e entrevistas. Os participantes responderam a um questionário composto de 15 sentenças em que expressaram suas opiniões categorizadas da seguinte forma: *Perceived Use* (PU) - 7

perguntas, *Perceived Easy of Use* (PEOU) - 4 perguntas e *Perceived Usage* (PUE) - 4 perguntas. Uma escala de *Likert* de 5 pontos foi proposta para medir as atitudes percebidas dos avaliadores, fornecendo uma gama de respostas a cada afirmação. A escala variou de (1) discordo totalmente, (2) discordo, (3) neutro, (4) concordo a (5) concordo fortemente.

Todos os avaliadores são profissionais da área de tecnologia da informação, trabalhando em empresas como programadores, analistas de sistemas, gerentes, diretores, professores assistentes e estudantes de pós-graduação. A ferramenta REQTRACE foi apresentada pessoalmente para cada avaliador. Ao final, cada avaliador respondeu ao questionário e os pontos sinalizados nas entrevistas foram catalogados.

Os dados coletados foram submetidos a uma análise descritiva e inferencial utilizando a frequência de contagem. Além disso, foi utilizada a técnica de correlação de Spearman para medir a correlação entre as variáveis (PU, PEOU, PUE). As seguintes hipóteses de avaliação da ferramenta foram consideradas:

- **H1.** A utilidade percebida influencia positivamente o uso percebido da ferramenta REQTRACE.
- **H2.** A facilidade de uso percebida influencia positivamente o uso percebido da ferramenta REQTRACE.
- **H3.** A facilidade de uso percebida influencia positivamente a utilidade percebida da ferramenta REQTRACE.

5.2 Avaliação Quantitativa - Questionário

Antes da aplicação do questionário foi realizada uma apresentação contemplando os conceitos de rastreabilidade de requisitos e SysML e o uso da ferramenta para cada avaliador. Cada apresentação teve uma duração média de 60 minutos. Inicialmente foram apresentados os conceitos de rastreabilidade de requisitos e SysML (20 minutos) e em seguida a apresentação dos requisitos a serem cadastrados na ferramenta REQTRACE (10 minutos). Foi utilizado um subconjunto de requisitos de um sistema de gerenciamento de trânsito (SOARES; VRANCKEN, 2008). Esta lista de requisitos é um trabalho de uma auditoria técnica com base em um estudo de literatura e entrevistas. Por fim, cada avaliador realizou o cadastro dos requisitos e das suas relações na ferramenta REQTRACE (30 minutos), assim como as consultas dos requisitos cadastrados.

Em seguida, cada avaliador respondeu o questionário baseado no modelo TAM. O questionário foi respondido por 24 profissionais da área de computação. As respostas a cada sentença são mostradas nas tabelas 5.1, 5.2 e 5.3, onde “pos” indica o número de respostas positivas. Foi considerado arbitrariamente como respostas positivas as respostas “concordo” ou

“concordo totalmente” (valores 4 ou 5). Para respostas negativas foram considerados os valores 1 e 2.

Tabela 5.1 – *Perceived usefulness* (PU) - Sentença 1 até 7 (N=24)

Sentença	1	2	3	4	5	pos
1 - O uso da ferramenta melhora o desempenho das atividades do gerenciamento de requisitos	0	0	0	10	14	24
2 - A informação fornecida na ferramenta é interessante para mim	0	0	4	9	11	20
3 - Eu acho que a ferramenta agrega valor	0	0	1	7	16	23
4 - Eu acho que a ferramenta permite melhor controle das mudanças nos requisitos	0	1	1	2	20	22
5 - Os atributos do cadastro de requisitos da ferramenta são importantes para o controle dos requisitos	0	0	1	7	16	23
6 - As relações entre os requisitos da ferramenta são importantes para o entendimento do impacto das mudanças nos requisitos	0	1	1	2	20	22
7 - O histórico de mudanças nos requisitos da ferramenta é importante durante o desenvolvimento do software	0	0	1	7	16	23

Tabela 5.2 – *Perceived ease of use* (PEOU) - Sentença 1 até 4 (N=24)

Sentença	1	2	3	4	5	pos
1 - A ferramenta é fácil de navegar	0	0	2	4	18	22
2 - Eu acho a ferramenta amigável	0	0	6	6	12	18
3 - Eu encontro rapidamente a informação que necessito na ferramenta	0	0	1	11	12	23
4 - De modo geral, a ferramenta é fácil de usar	0	1	0	7	16	23

Ao analisar os dados apresentados nas tabelas 5.1, 5.2 e 5.3, verifica-se que os participantes da pesquisa mostraram alta concordância em relação à maioria das sentenças da pesquisa. Isso significa que eles consideram que a ferramenta é fácil de usar, é útil e que eles iriam usá-la com frequência.

A partir dos dados apresentados, conclui-se que 93,6 % dos usuários têm opinião positiva (4 ou superior) para a maioria do questionário. A avaliação positiva dos entrevistados mostrou uma participação de 95,2 % para PU, 89,6 % para PEOU e 94,8 % para PUE. Com estes resultados, nota-se que a tecnologia é bem recebida, eles acreditam que o software é fácil de usar e melhora o desempenho em seu trabalho.

Os itens que mostraram maior aceitação da pesquisa (4 ou superior) foram a sentença 1 (PU) e a sentença 3 (PUE). Os itens que mostraram uma aceitação menor foram a sentença 2 (PU) e a sentença 2 (PEOU).

Tabela 5.3 – *Perceived usage* (PUE) - Sentença 1 até 4 (N=24)

Sentença	1	2	3	4	5	pos
1 - O uso da ferramenta aumenta o desempenho do meu trabalho	0	0	2	16	6	22
2 - O uso da ferramenta melhora a comunicação com os <i>stakeholders</i>	0	0	1	6	17	23
3 - A notificação por <i>e-mail</i> melhora a comunicação com relação às mudanças que ocorrem durante o desenvolvimento do software	0	0	0	9	15	24
4 - Eu gostaria de usar a ferramenta diariamente nas atividades da ER	0	0	1	9	13	22

A significância estatística de um resultado é uma medida estimada do grau em que esse resultado é “verdadeiro” (no sentido de que realmente o que ocorre na população ou no sentido de “representatividade da população”). A significância estatística representa um índice decrescente da confiabilidade de um resultado. Quanto maior o valor de significância, menos se pode acreditar que a relação observada entre as variáveis na amostra é um indicador confiável da relação entre as respectivas variáveis na população. Em muitas áreas de pesquisa, o valor de 0,05 é habitualmente tratado como um “limite aceitável” de erro (FIELD, 2009).

O coeficiente de correlação de Spearman mede a intensidade da relação entre variáveis ordinais. Para cada hipótese, foram consideradas como variáveis ordinais, o número de respostas de cada nível da escala de *Likert*. O coeficiente de correlação de Spearman varia entre -1 e 1. Quanto mais próximo destes extremos, maior a associação entre as variáveis.

Desta forma, todas as hipóteses foram confirmadas, uma vez que atingiu menos de 0,05 no valor de significância estatística. Com relação ao coeficiente de correlação de Spearman, foi mostrada uma correlação moderada apresentando a existência de uma correlação significativa e positiva para todas as hipóteses, conforme apresentado na Tabela 5.4.

Tabela 5.4 – Correlação entre as hipóteses

Hipóteses	Coeficiente de Spearman	Significância
H1 - A utilidade percebida influencia positivamente o uso percebido da ferramenta REQTRACE	0,431	0,036
H2 - A facilidade de uso percebida influencia positivamente o uso percebido da ferramenta REQTRACE	0,414	0,044
H3 - A facilidade de uso percebida influencia positivamente a utilidade percebida da ferramenta REQTRACE	0,502	0,012

5.3 Avaliação Qualitativa - Entrevistas

Todos os 24 avaliadores que responderam ao questionário também foram entrevistados com intuito de coletar informações referentes à visão geral da ferramenta. Foram catalogados também o nome do avaliador, cargo, tipo de empresa onde trabalha (pública, privada), idade, sexo e o número de anos de experiência na área da computação. Durante a entrevista, foram realizadas as seguintes perguntas:

1. Qual a visão geral da ferramenta REQTRACE?
2. A ferramenta REQTRACE é realmente útil?
3. O que é desnecessário na ferramenta REQTRACE?
4. O que faltou na ferramenta REQTRACE?

Com relação à primeira pergunta, os entrevistados destacaram a importância da ferramenta por auxiliar o gerenciamento e a rastreabilidade dos requisitos e a importância da comunicação automática com os *stakeholders*. Para a maioria dos entrevistados, a ferramenta é simples e de fácil utilização. Entre outras respostas mencionadas nesta pergunta, destacaram-se: “Provê a integração entre os requisitos e o mais importante, a comunicação com os *stakeholders* de qualquer alteração nestes”, “Muito simples e de fácil navegação. Ferramenta útil para gerenciar requisitos e suas dependências”, “Bastante eficiente, simples e funcional. Atende e possibilita o mapeamento e gerenciamento de requisitos”, “Ferramenta intuitiva e fácil de utilizar”, “Muito interessante! Fiquei interessado em algo similar!”, “Muito boa e rápida”, “Bem interessante, simples e fácil de utilizar” e “Boa navegabilidade e *menus* sugestivos”.

Com relação à segunda pergunta, os entrevistados destacaram que a ferramenta é útil e ressaltaram a importância das relações SysML entre os requisitos e o monitoramento dos requisitos como as principais utilidades e vantagens da ferramenta REQTRACE. Entre outras respostas mencionadas na entrevista com relação à esta pergunta, destacaram-se: “Sim. É importante conhecer no detalhe os requisitos, bem como os seus relacionamentos para uma melhor gestão”, “Sim. Auxilia bastante durante o processo de gerenciamento de requisitos”, “Sim. Porque foram implementadas técnicas de gerenciamento de requisitos e rastreabilidade de requisitos”, “Com certeza. Ajuda bastante na rastreabilidade das alterações”, “Sim, pois facilita o gerenciamento dos requisitos e permite o registro dos relacionamentos entre eles e de todas as mudanças efetuadas”, “Sim. Permite um melhor controle das mudanças e relacionamentos entre os requisitos”, “Sim. Os requisitos são gerenciados e é possível visualizar a rastreabilidade entre os requisitos” e “Muito útil! Empolgante!”.

Com relação ao que é desnecessário na ferramenta, 100 % dos entrevistados não identificaram nenhum atributo e funcionalidade desnecessários. Entre outras respostas, destacaram-se: “Todas as funcionalidades apresentadas mostraram-se necessárias”, “Não encontrei nenhuma

funcionalidade desnecessária”, “Não se aplica”, “Nada”, “Nada. A ferramenta propõe o rastreamento básico e suficiente para o gerenciamento”, “Nada. Todas as informações são importantes” e “Não encontrei nada que seja desnecessário”.

Com relação a última pergunta, os entrevistados sugeriram diversas melhorias na ferramenta. Entre as sugestões mais citadas, destacam-se: melhorias nas consultas, principalmente no que se refere a novos filtros com o intuito de facilitar as pesquisas, adição de novas opções no atributo status do requisito no cadastro de requisitos (requisito em fase de teste, requisito em homologação, requisito entregue (encerrado, concluído)), adição da data esperada de entrega do requisito no cadastro de requisitos para controlar as datas previstas de conclusão dos requisitos, adição de novas informações no *e-mail* enviado aos *stakeholders* (atributo modificado e o novo valor do atributo), criar funcionalidade para visualização em diagramas das relações entre os requisitos (relações SysML) e disponibilizar a ferramenta tanto para ambientes Web como para dispositivos móveis.

Entre outras respostas mencionadas na última pergunta, destacaram-se: “Possibilidade de gerar diagramas/gráficos das rastreabilidades e seus relacionamentos”, “Incluir filtros na consulta de requisitos; incluir um atributo para informar a data prevista da implantação do requisito; incluir status de encerramento do requisito”, “A ferramenta poderia ser web e/ou mobile”, “Ter mais opções de status do requisito”, “A lista de opções de status do requisito pode ser enriquecida”, “Implementar visualmente (tipo um diagrama de classes) o relacionamento entre os requisitos”, “Visão gráfica/diagrama que mostre as relações entre os requisitos e seus impactos”, “Uma interface web ou mobile” e “O campo status do requisito não tem a possibilidade de modificar para concluído; permitir geração de diagrama de requisitos com essas ligações e dependências (de preferência, web)”.

Diversos entrevistados afirmaram que utilizariam de forma imediata a ferramenta em suas atividades diárias e destacaram a ferramenta como “muito interessante”, “funcional” e “direta”.

A Tabela 5.5 apresenta o detalhamento das informações catalogadas durante a entrevista e o perfil dos entrevistados.

Com relação aos cargos atuais dos entrevistados, 50,0 % são analistas de sistemas e programadores, 25,0 % ocupam cargos de gestão (diretoria e gerência), 20,8 % são professores e 4,2 % são estudantes de pós-graduação. O cargo do entrevistado refere-se ao cargo atual, sendo que muitos deles possuem experiência em mais de uma instituição.

Analisando o tipo de empresa de cada entrevistado, 58,3 % são funcionários públicos e 41,7 % estão trabalhando em instituições privadas. Com relação aos funcionários públicos, eles estão alocados na esfera Estadual e Federal. Os 24 avaliadores estão distribuídos em 05 empresas públicas e em 05 instituições privadas.

Tabela 5.5 – Dados coletados na entrevista

Avaliador	Cargo	Tipo de Empresa	Idade	Sexo	Anos de Experiência em TI
01	Gerente de Projeto	Privada	40	M	21
02	Diretor Comercial - TI	Privada	37	M	19
03	Analista de Sistemas	Privada	33	F	11
04	Diretor de Operações- TI	Privada	36	M	15
05	Analista de Sistemas	Pública	34	M	12
06	Analista de Sistemas	Pública	34	M	14
07	Analista de Sistemas	Privada	33	M	14
08	Professor	Pública	40	F	20
09	Estudante de Mestrado	Pública	27	F	0
10	Professor	Pública	35	F	10
11	Analista de Sistemas	Pública	42	M	15
12	Gerente de Projeto	Pública	35	M	15
13	Analista de Sistemas	Pública	39	F	15
14	Diretor de Operações - TI	Privada	35	M	15
15	Professor	Pública	42	M	12
16	Analista de Sistemas	Pública	38	M	20
17	Analista de Sistemas	Privada	39	M	17
18	Analista de Sistemas	Pública	34	M	10
19	Analista de Sistemas	Pública	37	M	20
20	Professor	Privada	42	M	20
21	Analista de Sistemas	Privada	45	M	27
22	Gerente de Sistemas	Privada	34	M	18
23	Professor	Pública	40	M	18
24	Programador	Privada	28	M	10

Com relação à idade dos entrevistados, 8,3 % possuem até 30 anos, 62,5 % possuem entre 31 e 39 anos e 29,2 % possuem 40 anos ou mais.

Analisando o sexo dos entrevistados, 79,2 % são do sexo masculino e 20,8 % do sexo feminino. Analisando o sexo e o cargo dos entrevistados em conjunto, temos 100,0 % dos que ocupam cargos de gestão são do sexo masculino, 80,0 % das mulheres entrevistadas estão atuando em instituições públicas e 40,0 % dos professores entrevistados são do sexo feminino.

Com relação ao número de anos de experiência na área de computação, 4,2 % tem até 5 anos de experiência, 12,5 % entre 6 e 10 anos de experiência, 41,7 % entre 11 e 15 anos de experiência, 33,3 % entre 16 e 20 anos de experiência e 8,3 % com 21 anos ou mais de experiência.

Analisando o questionário e as entrevistas em conjunto foi observado que as rejeições em algumas sentenças (notas 1 e 2) no questionário aplicado foram atribuídas pelo único entrevistado com cargo de programador. A falta de experiência em atividades de análise e gestão de sistemas e o direcionamento das atividades de um programador para a implementação do software

podem contribuir para a falta de visão e importância da técnica e da ferramenta propostas nesta dissertação. Por sua vez, os demais entrevistados (analista de sistemas, gerentes, diretores e professores) não atribuíram notas 1 e 2, destacando a relevância das propostas.

5.4 Conclusão do Capítulo

Este capítulo teve como objetivo avaliar a técnica de rastreabilidade entre requisitos apoiada pela ferramenta REQTRACE. Foram utilizados critérios de avaliação baseados no modelo TAM (método de avaliação empírico aplicado para entender o uso e comportamento dos usuários de uma tecnologia). Para a avaliação foram utilizadas as técnicas de questionário e entrevistas.

Houve grande aceitação da proposta da técnica e da ferramenta REQTRACE sendo avaliadas por diversos profissionais com experiência em análise, modelagem, projeto, implementação e implantação de grandes sistemas legados e *Enterprise Resource Planning* (ERP).

O perfil dos avaliadores contemplou funcionários públicos e de instituições privadas tornando uma avaliação robusta com pessoas de visões, opiniões e experiências diversificadas. A experiência dos avaliadores em diversos tipos de sistema, como área comercial, logística, financeira, contabilidade, fiscal, marketing, acadêmica, jurídica reforça que as propostas deste trabalho podem ser implementadas na indústria sem grandes dificuldades e desafios.

Foi percebido também que muitos avaliadores desconheciam a linguagem SysML e foi notado interesses em utilizar o diagrama de Requisitos em suas atividades diárias.

Com relação às hipóteses sobre o uso da ferramenta REQTRACE, todas foram confirmadas, ou seja, a utilidade percebida influenciou positivamente o uso percebido, a facilidade de uso percebida influenciou positivamente o uso percebido e a facilidade de uso percebida influenciou positivamente a utilidade percebida.

No Capítulo 6 são descritas as considerações finais do trabalho, apresentando a conclusão da dissertação, as principais contribuições, ameaças à validade e os trabalhos futuros.

6 Conclusão

Alguns dos grandes desafios nos projetos de software estão relacionados à redução de custos e ao cumprimento dos prazos, bem como fazer com que o produto a ser desenvolvido atenda a todas as necessidades dos clientes. A rastreabilidade de requisitos faz parte desse cenário visando buscar soluções para estes desafios.

A implementação da rastreabilidade de requisitos ainda é repleta de problemas e barreiras. Os problemas podem ser categorizados em problemas de gerenciamento, sociais e técnicos. Este trabalho direcionou suas pesquisas a um dos problemas sociais relativos à falta de comunicação entre as equipes, com foco nos *stakeholders*, e em um dos problemas técnicos que se refere a falta de utilização, na prática, de técnicas e ferramentas.

No Capítulo 3, foi realizada uma revisão de técnicas de rastreabilidade de requisitos e uma análise comparativa de ferramentas a partir de um conjunto de critérios propostos. A fonte para esta avaliação foi uma pesquisa na Internet sobre as principais técnicas e ferramentas. Estas pesquisas correspondem aos objetivos específicos 1 e 2 da dissertação.

As técnicas de rastreabilidade de requisitos se complementam e podem ser combinadas, ou seja, em um projeto pode ser aplicada mais de uma técnica aproveitando as vantagens de cada uma. A maioria das técnicas pesquisadas não foi validada empiricamente, tornando, principalmente para a indústria, a sua utilização e aplicação mais difícil.

As ferramentas avaliadas não dispõem de um tratamento automático no processo de alteração dos atributos dos requisitos. Como exemplo, uma mensagem automática pode ser enviada aos *stakeholders* quando um atributo de um requisito é modificado. Esta limitação foi abordada na proposta principal desta dissertação.

No Capítulo 4 foi proposta uma técnica de rastreabilidade entre requisitos utilizando a técnica baseada em evento permitindo a comunicação automática com *stakeholders*. Na técnica proposta são utilizadas as relações e as tabelas SysML descritas na Seção 2.6. A técnica proposta refere-se ao objetivo específico 3 da dissertação. Neste mesmo capítulo foi apresentada a ferramenta REQTRACE que apoia a técnica proposta.

No Capítulo 5 foi apresentada a avaliação da técnica e da ferramenta utilizando questionário, baseado no modelo TAM aplicados para entender o uso e comportamento dos usuários em uma tecnologia, e entrevistas. Estas avaliações correspondem ao objetivo específico 4.

Houve ampla aceitação da proposta da técnica e da ferramenta REQTRACE sendo avaliadas por profissionais com experiência em todas etapas do ciclo de vida do software. O perfil dos avaliadores contemplou funcionários públicos e de instituições privadas, avaliadores com conhecimento em sistemas legados e ERP e com experiências diversificadas.

A experiência dos avaliadores em diversos tipos de sistema, como área comercial, logística, financeira, marketing, acadêmica e jurídica, reforça que as propostas deste trabalho podem ser implementadas na indústria sem grandes dificuldades e desafios e que a técnica e a ferramenta propostas são genéricas.

Como a SysML é uma linguagem baseada na UML, a utilização de conceitos da SysML (relações e tabelas) para equipes que conhecem UML tende a ser realizada com menos esforço, pois a UML é utilizada na academia e na indústria.

Finalmente, é possível observar que as propostas sugeridas são pertinentes no contexto da rastreabilidade de requisitos e que pode ser utilizada independente do tipo de sistema contemplando melhorias nas atividades diárias dos profissionais de computação, além de sua fácil utilização e aplicação.

6.1 Contribuições do Trabalho

Esta dissertação apresentou diversas contribuições para a pesquisa e prática da Engenharia de Requisitos, especialmente para a subárea de Rastreabilidade de Requisitos.

A primeira contribuição foi a pesquisa do estado da arte em rastreabilidade de requisitos que contribuiu para entender o que está sendo pesquisado nesta área. Esta pesquisa foi fundamental para compreender a amplitude do que vem sendo produzido pela comunidade científica e contribuiu para as propostas principais desta dissertação. Esta contribuição foi descrita na Seção 1.4.

Outra contribuição foi a realização de uma análise comparativa das principais técnicas de rastreabilidade de requisitos. Esta análise é importante para identificar as vantagens e desvantagens de cada técnica. A contribuição a este item foi mostrada na Seção 3.2, com a comparação de cinco técnicas de rastreabilidade de requisitos.

Uma análise comparativa de cinco ferramentas de rastreabilidade de requisitos contendo nove critérios de avaliação também foi uma contribuição deste trabalho. Esta análise comparativa serviu para delinear a pesquisa principalmente na criação da técnica proposta e da ferramenta REQTRACE. A contribuição a este item foi mostrada na Seção 3.5.

A principal contribuição do trabalho foi a proposta de uma técnica de rastreabilidade entre requisitos utilizando rastreabilidade baseada em evento combinada com SysML e apoiada pela ferramenta REQTRACE. Foi escolhida a rastreabilidade entre requisitos por ser pouco estudada conforme mencionado na Seção 1.4. A contribuição a este item foi mostrada no Capítulo 4.

Por fim, outra contribuição foi a avaliação da técnica e da ferramenta propostas. A avaliação foi realizada por vinte e quatro profissionais da área de computação com ampla aceitação das propostas. Esta contribuição foi mostrada no Capítulo 5.

Esta pesquisa também resultou em dois artigos em revistas e congressos científicos, são eles:

JESUS, T. O, D. SOARES, M. S. “*A Multi-Criteria Analysis of Techniques and Tools for Tracing Software Requirements*” publicado na Revista IEEE América Latina (Maio, 2017).

JESUS, T. O, D. SOARES, M. S. “*An Event-Based Technique to Trace Requirements Modeled with SysML*” publicado nas *Proceedings da 17th International Conference on Computational Science and Its Applications* (ICCSA, 2017).

6.2 Ameaças à Validade

A validade de uma pesquisa se refere à capacidade que os métodos utilizados na pesquisa propiciam à consecução fidedigna de seus objetivos. A validade de uma investigação é, normalmente, dividida em validade interna e externa. A primeira é relativa à coerência entre as conclusões de um estudo e a realidade. A segunda está ligada à generalização dos resultados e à possibilidade de, a partir das conclusões de um estudo, se poder aplicar a outro grupo alvo (replicabilidade).

A validade interna dá mais ênfase à validade do processo ou do método aplicado. Neste trabalho, com relação aos procedimentos metodológicos poderia ser realizada uma revisão sistemática de rastreabilidade entre requisitos com o objetivo de resumir toda a informação sobre o tema de maneira formal, completa e minuciosa.

A validade externa de uma pesquisa se refere à capacidade de generalizar os resultados, obtidos com a amostra, à população em geral. Alguns vieses, quando não controlados, provocam limitação da validade externa da pesquisa, ou seja, conduzem a uma grande especificidade dos resultados obtidos àquele tipo de sujeito estudado ou ao contexto específico da pesquisa ou a algum conjunto indesejável de condições específicas daquele experimento. Para aumentar a validade externa desta pesquisa, foram selecionados avaliadores de diversos cargos, empresas, tipos de empresa, experiências e faixas etárias com o intuito de garantir que as conclusões da pesquisa são verdadeiras para outros contextos e outras pessoas.

6.3 Limitações da Pesquisa

Embora os resultados deste trabalho tenham se mostrado satisfatórios, ao desenvolver este trabalho, foram encontradas algumas limitações:

- Foram analisadas apenas cinco ferramentas de rastreabilidade de requisitos na Seção 3.5. Além disto, esta avaliação foi realizada apenas por uma pessoa. Existem diversas ferramentas de rastreabilidade de requisitos, mas não foi possível avaliar dezenas delas

pois algumas não funcionaram quando foram instaladas conforme mencionado na Seção 3.2. A avaliação foi realizada somente por uma pessoa em virtude da dificuldade de encontrar profissionais com disponibilidade para instalação, utilização e análise das cinco ferramentas.

- Na ferramenta REQTRACE, as mensagens de notificações das mudanças nos requisitos são enviadas somente se no momento da modificação tiver conexão com a Internet. Não foi implementada nenhuma funcionalidade quando a Internet estiver *off-line*.
- Foram utilizadas vinte e quatro pessoas na avaliação da técnica e da ferramenta propostas. Não se trata de uma quantidade baixa, mas para uma validação efetiva são necessários mais avaliadores. Para mais análises estatísticas, este número é considerado baixo.

6.4 Trabalhos Futuros

Para continuidade e aprimoramento desta dissertação, os seguintes trabalhos futuros podem ser realizados.

- Sugere-se melhorar a ferramenta REQTRACE com a implementação das sugestões dadas pelos avaliadores mencionadas na Seção 5.3.
- Aplicar a técnica e a ferramenta propostas em um estudo de caso mensurando as dificuldades, vantagens e desvantagens de seu uso. A utilização da ferramenta em uma empresa testará a escalabilidade e a confiabilidade da ferramenta, pois serão cadastrados centenas de requisitos e estes serão modificados constantemente por dezenas de usuários.
- Solicitar o registro da ferramenta REQTRACE.
- Novas pesquisas sobre o uso de outras técnicas de rastreabilidade de requisitos combinadas com UML e SysML podem ser realizadas.

Referências

- ADAMS, D. A.; NELSON, R. R.; TODD, P. A. Perceived Usefulness, Ease of Use, and Usage of Information Technology: A Replication. *MIS Quarterly*, JSTOR, p. 227–247, 1992.
- ARKLEY, P.; RIDDLE, S. Overcoming the Traceability Benefit Problem. In: *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on*. Paris: IEEE, 2005. p. 385–389.
- ASUNCION, H. U.; FRANCOIS, F.; TAYLOR, R. N. An End-to-End Industrial Software Traceability Tool. In: *Proceedings of the the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering*. New York, NY, USA: ACM, 2007. (ESEC-FSE '07), p. 115–124.
- BADREDDIN, O.; STURM, A.; LETHBRIDGE, T. C. Requirement Traceability: A Model-Based Approach. In: IEEE. *Model-Driven Requirements Engineering Workshop (MoDRE), 2014 IEEE 4th International*. Karlskrona, 2014. p. 87–91.
- BALMELLI, L. An Overview of the Systems Modeling Language for Products and Systems Development. *Journal of Object Technology*, v. 6, n. 6, p. 149–177, 2007.
- BALMELLI, L.; BROWN, D.; CANTOR, M.; MOTT, M. Model-Driven Systems Development. *IBM Systems Journal*, IBM, v. 45, n. 3, p. 569–585, 2006.
- BIRÓ, M.; KLESPLITZ, J.; GMEINER, J.; ILLIBAUER, C.; KOVÁCS, L. Towards Automated Traceability Assessment through Augmented Lifecycle Space. In: SPRINGER. *European Conference on Software Process Improvement*. Austria, 2016. p. 94–105.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: Guia do Usuário*. 6. ed. Brasil: Elsevier Brasil, 2006.
- BORG, M.; RUNESON, P.; BRODÉN, L. Evaluation of Traceability Recovery in Context: A Taxonomy for Information Retrieval Tools. IET, 2012.
- BRIAND, L.; FALESSI, D.; NEJATI, S.; SABETZADEH, M.; YUE, T. Traceability and SysML Design Slices to Support Safety Inspections: A Controlled Experiment. *ACM Transactions on Software Engineering Methodology*, v. 23, n. 1, p. 9:1–9:43, feb 2014.
- CHANG, C.-H.; LU, C.-W.; YANG, W. P.; CHU, W. C.-C.; YANG, C.-T.; TSAI, C.-T.; HSIUNG, P.-A. A SysML Based Requirement Modeling Automatic Transformation Approach. In: IEEE. *Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE 38th International*. Vasteras, 2014. p. 474–479.
- CHARRADA, E. B.; KOZIOLEK, A.; GLINZ, M. Identifying Outdated Requirements Based on Source Code Changes. In: *Requirements Engineering Conference (RE), 2012 20th IEEE International*. Chicago: IEEE, 2012. p. 61–70.
- CLELAND-HUANG, J. Toward Improved Traceability of Non-Functional Requirements. In: *Proceedings of the 3rd International Workshop on Traceability in Emerging Forms of Software Engineering*. New York, NY, USA: ACM, 2005. p. 14–19.

- CLELAND-HUANG, J.; CHANG, C. K.; CHRISTENSEN, M. Event-Based Traceability For Managing Evolutionary Change. *IEEE Transactions on Software Engineering*, IEEE, v. 29, n. 9, p. 796–810, 2003.
- CLELAND-HUANG, J.; CHANG, C. K.; SETHI, G.; JAVVAJI, K.; HU, H.; XIA, J. Automating Speculative Queries through Event-Based Requirements Traceability. In: IEEE. *Proceedings of the IEEE Joint International Conference on Requirements Engineering*. Essen, 2002. p. 289–296.
- CLELAND-HUANG, J.; GOTEL, O.; ZISMAN, A. *Software and Systems Traceability*. GBR: Springer, 2012.
- CLELAND-HUANG, J.; GOTEL, O. C.; HAYES, J. H.; MÄDER, P.; ZISMAN, A. Software Traceability: Trends and Future Directions. In: *Proceedings of the on Future of Software Engineering*. Hyderabad, India: ACM, 2014. p. 55–69.
- CORDEIRO, A.; FREITAS, A. L. Priorização de Requisitos e Avaliação da Qualidade de Software Segundo a Percepção dos Usuários. *Ciência da Informação*, I, n. 2, p. 160–179, 2012.
- CYSNEIROS FILHO, G. A. de A. Software Traceability for Multi-Agent Systems Implemented Using BDI Architecture. *DPhil Thesis*, London, 2011.
- DAVIS, F. D. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, JSTOR, p. 319–340, 1989.
- EDER, S.; FEMMER, H.; HAUPTMANN, B.; JUNKER, M. Configuring Latent Semantic Indexing for Requirements Tracing. In: IEEE PRESS. *Proceedings of the Second International Workshop on Requirements Engineering and Testing*. Piscataway, 2015. p. 27–33.
- EGYED, A.; GRÜNBACHER, P.; HEINDL, M.; BIFFL, S. Value-Based Requirements Traceability: Lessons Learned. In: *Design Requirements Engineering: A Ten-Year Perspective*. New York: Springer, 2009. p. 240–257.
- EUGSTER, P. T.; FELBER, P. A.; GUERRAOUI, R.; KERMARREC, A.-M. The Many Faces of Publish/Subscribe. *ACM Computing Surveys (CSUR)*, ACM, v. 35, n. 2, p. 114–131, 2003.
- FACHIN, O. *Fundamentos de Metodologia*. São Paulo: Saraiva, 2005.
- FIELD, A. *Discovering Statistics Using SPSS*. Los Angeles: Sage publications, 2009.
- FRIEDENTHAL, S.; MOORE, A.; STEINER, R. *A Practical Guide to SysML: The Systems Modeling Language*. 3. ed. Waltham: Morgan Kaufmann, 2014.
- GIL, A. C. *Métodos e Técnicas de Pesquisa Social*. 6. ed. São Paulo: Editora Atlas SA, 2008.
- GUPTA, S.; HARTKOPF, J. M.; RAMASWAMY, S. Event Notifier: A Pattern for Event Notification. In: CAMBRIDGE UNIVERSITY PRESS. *More Java Gems*. New York, 2000. p. 131–153.
- HAMMER, T.; ROSENBERG, L.; HUFFMAN, L.; HYATT, L. Requirement Metrics-Value Added. In: *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*. USA: IEEE, 1997. p. 141.

- HEINDL, M.; BIFFL, S. A Case Study on Value-Based Requirements Tracing. In: *Proceedings of the 10th European Software Engineering Conference Held Jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. New York: ACM, 2005. p. 60–69.
- HONG, Y.; KIM, M.; LEE, S.-W. Requirements Management Tool With Evolving Traceability For Heterogeneous Artifacts in the Entire Life Cycle. In: *Software Engineering Research, Management and Applications (SERA), 2010 Eighth ACIS International Conference on*. Washington: IEEE, 2010. p. 248–255.
- HUANG, J. *Robust Requirements Traceability for Handling Evolutionary and Speculative Change*. Chicago: University of Illinois at Chicago, 2002.
- KANNENBERG, A.; SAIEDIAN, H. Why Software Requirements Traceability Remains a Challenge. *The Journal of Defense Software Engineering*, v. 22, n. 5, p. 14–19, 2009.
- LAGO, P.; MUCCINI, H.; VLIET, H. V. A Scoped Approach to Traceability Management. *Journal of Systems and Software*, Elsevier, v. 82, n. 1, p. 168–182, 2009.
- LUCIA, A. D.; FASANO, F.; OLIVETO, R.; TORTORA, G. Recovering Traceability Links in Software Artifact Management Systems Using Information Retrieval Methods. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, ACM, v. 16, n. 4, p. 13, 2007.
- MÄDER, P.; GOTEL, O.; PHILIPPOW, I. Rule-Based Maintenance of Post-Requirements Traceability Relations. In: *IEEE. International Requirements Engineering, 2008. RE'08. 16th IEEE*. Barcelona, 2008. p. 23–32.
- MARCUS, A.; MALETIC, J. I.; SERGEYEV, A. Recovery of Traceability Links Between Software Documentation and Source Code. *International Journal of Software Engineering and Knowledge Engineering*, World Scientific, v. 15, n. 05, p. 811–836, 2005.
- MARQUES, M. R. S.; SIEGERT, E.; BRISOLARA, L. Integrating UML, MARTE and SysML to Improve Requirements Specification and Traceability in the Embedded Domain. In: *IEEE. Industrial Informatics (INDIN), 2014 12th IEEE International Conference on*. Porto Alegre, 2014. p. 176–181.
- MIN, H.-S. Traceability Guideline for Software Requirements and UML Design. *International Journal of Software Engineering and Knowledge Engineering*, World Scientific, v. 26, n. 01, p. 87–113, 2016.
- NAIR, S.; VARA, J. L. de la; SEN, S. A Review of Traceability Research at the Requirements Engineering Conference. In: *Requirements Engineering Conference (RE), 2013 21st IEEE International*. Rio de Janeiro: IEEE, 2013. p. 222–229.
- NIU, N.; MAHMOUD, A. Enhancing Candidate Link Generation for Requirements Tracing: The Cluster Hypothesis Revisited. In: *2012 20th IEEE International Requirements Engineering Conference (RE), Chicago, IL, USA, September 24-28, 2012*. USA: IEEE, 2012. p. 81–90.
- OH, J.; KANG, S. A Hierarchical Model for Traceability Between Requirements and Architecture. In: *ACM. Proceedings of the 29th Annual ACM Symposium on Applied Computing*. New York, 2014. p. 1035–1042.

- PARVIAINEN, P.; TIHINEN, M.; LORMANMS, M.; SOLINGEN, R. van. *Requirements Engineering for Sociotechnical Systems*. Reino Unido: Idea Group Inc., 2005. 1–20 p.
- PFLEEGER, S. L. *Engenharia de Software: Teoria e Prática*. Brasil: Prentice Hall Brasil, 2004.
- PINHEIRO, F. A. Requirements Traceability. *Kluwer International Series in Engineering and Computer Science*, Springer, p. 91–114, 2004.
- PLATANIOTIS, G.; MA, Q.; PROPER, E.; KINDEREN, S. de. Traceability and Modeling of Requirements in Enterprise Architecture from a Design Rationale Perspective. In: IEEE. *Research Challenges in Information Science (RCIS), 2015 IEEE 9th International Conference on*. Piscataway, 2015. p. 518–519.
- POHL, K.; RUPP, C. *Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam-Foundation Level-IREB Compliant*. Exeter, United Kingdom: Rocky Nook, Inc., 2011.
- PRESSMAN, R. S. *Engenharia de software*. 8. ed. Brasil: McGraw Hill Brasil, 2016.
- RAHMAN, M. M.; RIPON, S. Elicitation and Modeling Non-Functional Requirements - A POS Case Study. *CoRR*, abs/1403.1936, 2014.
- REGAN, G.; MCCAFFERY, F.; MCDAID, K.; FLOOD, D. The Barriers to Traceability and Their Potential Solutions: Towards a Reference Framework. In: *Software Engineering and Advanced Applications (SEAA), 2012 38th Euromicro Conference on*. Turkey: IEEE, 2012. p. 319–322.
- RIBEIRO, F. G. C.; RETTBERG, A.; PEREIRA, C. E.; SOARES, M. S. A Model-Based Engineering Methodology for Requirements and Formal Design of Embedded and Real-Time Systems. In: *Proceedings of the Hawaii International Conference on System Sciences (HICSS)*. Honolulu: IEEE, 2017. p. 6131–6140.
- ROBERTSON, S.; ROBERTSON, J. *Mastering the Requirements Process*. 3. ed. USA: Addison-Wesley Professional, 2006.
- ROCHIMAH, S.; WAN-KADIR, W. M.; ABDULLAH, A. H. An Evaluation of Traceability Approaches to Support Software Evolution. In: *ICSEA*. Cap Esterel: IEEE, 2007. p. 19.
- ROMANOWSKI, J. P.; ENS, R. T. As Pesquisas Denominadas do Tipo Estado da Arte em Educação. *Diálogo Educacional*, v. 6, n. 19, p. 37–50, 2006.
- SCHNEIDER, F.; NAUGHTON, H.; BERENBACH, B. A Modeling Language to Support Early Lifecycle Requirements Modeling for Systems Engineering. *Procedia Computer Science*, Elsevier, v. 8, p. 201–206, 2012.
- SHAW, M. Writing Good Software Engineering Research Papers. In: IEEE. *Software Engineering, 2003. Proceedings. 25th International Conference on*. Portland, 2003. p. 726–736.
- SJOBERG, D. I.; DYBA, T.; JORGENSEN, M. The Future of Empirical Methods in Software Engineering Research. In: *2007 Future of Software Engineering*. USA: IEEE Computer Society, 2007. p. 358–378.
- SOARES, M. S.; CIOQUETTA, D. S. Analysis of Techniques for Documenting User Requirements. In: *Computational Science and Its Applications-ICCSA 2012*. Brasil: Springer, 2012. p. 16–28.

- SOARES, M. S.; VRANCKEN, J. Model-Driven User Requirements Specification Using SysML. *Journal of Software*, v. 3, n. 6, p. 57–68, 2008.
- SOARES, M. S.; VRANCKEN, J.; VERBRAECK, A. User Requirements Modeling and Analysis of Software-Intensive Systems. *Journal of Systems and Software*, Elsevier, v. 84, n. 2, p. 328–339, 2011.
- SOMMERVILLE, I. *Software Engineering*. 10. ed. USA: Pearson, 2015.
- SOMMERVILLE, I.; KOTONYA, G. *Requirements Engineering: Processes and Techniques*. USA: John Wiley & Sons, Inc., 1998.
- SPANOUDAKIS, G.; ZISMAN, A.; PÉREZ-MINANA, E.; KRAUSE, P. Rule-Based Generation of Requirements Traceability Relations. *Journal of Systems and Software*, Elsevier, v. 72, n. 2, p. 105–127, 2004.
- THAYER, R.; DORFMAN, M. *Software Requirements Engineering Glossary*. Wiley-IEEE Press, 1997.
- THOMMAZO, A. D.; MALIMPENSA, G.; OLIVEIRA, T. R. de; OLIVATTO, G.; FABBRI, S. C. Requirements Traceability Matrix: Automatic Generation and Visualization. In: *IEEE Software Engineering (SBES), 2012 26th Brazilian Symposium on*. Brasil, 2012. p. 101–110.
- TOGNERI, D. F.; FALBO, R. de A.; MENEZES, C. S. de. Supporting Cooperative Requirements Engineering with an Automated Tool. In: *Anais do WER02 - Workshop em Engenharia de Requisitos, Valencia, Espanha, Novembro 11-12, 2002*. Valencia, Espanha: WER, 2002. p. 240–254.
- TORKAR, R.; GORSCHKE, T.; FELDT, R.; SVAHNBERG, M.; RAJA, U. A.; KAMRAN, K. Requirements Traceability: a Systematic Review and Industry Case Study. *International Journal of Software Engineering and Knowledge Engineering*, World Scientific, v. 22, n. 03, p. 385–433, 2012.
- VENTURA, P. *Ebook: Requisitos de Software*. 2013.
- WIEGERS, K. *Software Requirements: Practical Techniques for Gathering and Managing Requirements Throughout the Product Development Cycle*. 2. ed. USA: Microsoft Press, 2003. (Developer Best Practices Series).
- WOHLIN, C.; AURUM, A. *Engineering and Managing Software Requirements*. USA: Springer Science & Business Media, 2005. 478 p.
- YU, A. T.; SHEN, G. Q. Problems and Solutions of Requirements Management for Construction Projects under the Traditional Procurement Systems. *Facilities*, Emerald Group Publishing Limited, v. 31, n. 5/6, p. 223–237, 2013.